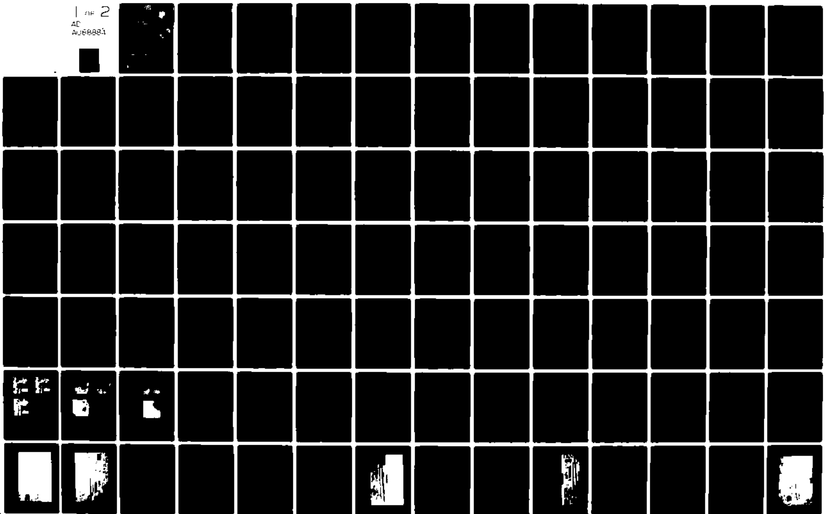
AD-A088 884   PATTERN ANALYSIS AND RECOGNITION CORP ROME NY   F/G 9/2
IMAGE FEATURE MANUSCRIPT GENERATION.(U)
MAY 80   M J ZORACKI, D A MARKS, J L CAMBIER   F30602-78-C-0017
UNCLASSIFIED   PAR-80-25   RADC-TR-80-176   NL

1 OF 2
AD
A088884



LEVEL

RADC-TR-80-176
Final Technical Report
May 1980

AD A088884

IMAGE FEATURE MANUSCRIPT GENERATION

Pattern Analysis and Recognition Corporation

Dr. Michael J. Zoracki
Mr. David A. Marks
Dr. James L. Cambier

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

ORIGINAL CONTAINS COLOR PLATES: ALL DDC REPRODUCTIONS WILL BE IN BLACK AND WHITE.

80 9 8 084

This report has been reviewed by the RADC Public Affairs [illegible] and is releasable to the National Technical Information Service [illegible] At NTIS it will be releasable to the general public, including [illegible] nations.

RADC-TR-80-176 has been reviewed and is approved for publication.

APPROVED: Frederick W. Rahrig

FREDERICK W. RAHRIG
Project Engineer

APPROVED: Thadeus J. Domurat

THADEUS J. DOMURAT, Acting Chief
Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRE), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-80-176	2. GOVT ACCESSION NO. AD-A088 884	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) IMAGE FEATURE MANUSCRIPT GENERATION		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report
		6. PERFORMING ORG. REPORT NUMBER PAR Report #80-25
7. AUTHOR(s) Dr. Michael J. Zoracki Mr. David A. Marks Dr. James L. Cambier		8. CONTRACT OR GRANT NUMBER(s) F30602-78-C-0017
9. PERFORMING ORGANIZATION NAME AND ADDRESS Pattern Analysis and Recognition Corporation 228 Liberty Plaza Rome NY 13440		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 45941714
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRRE) Griffiss AFB NY 13441		12. REPORT DATE May 1980
		13. NUMBER OF PAGES 161
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Frederick W. Rahrig (IRRE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pattern Recognition
Feature Extraction
Image Classification
Image Segmentation
Atmospheric Modeling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the Image Feature Manuscript Generation capability developed under RADC Contract F30602-78-C-0017. The capability developed here will serve as an experimental facility on which to evaluate advanced techniques of potential application to the Automatic Feature Extraction System (AFES).

This effort has been directed at providing an improved interactive means of classifying photographic imagery and for addressing and (Cont'd)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

390101

Item 20 (Cont'd)

displaying graphics for the purpose of automatically categorizing image features, generating feature boundaries, and editing the shapes and locations of those boundaries.

Specific attention was given to the development of a region growing capability and to the inclusion of an AFAL-developed atmospheric model. These items are described at length within this report.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

EVALUATION

This report describes the development of an image processing capability to classify a photographic image into a number of categories based on preprogrammed feature extraction parameters. An interactive graphics capability was also developed to correct and convert these category boundaries into graphic vector files suitable for output on an existing plotter and for tabulation in a feature analysis data table. An atmospheric model was also integrated into this system which should prove useful in accomplishing the goals of technical planning objective, Reconnaissance and Intelligence (TPO 2) in allowing for signature extension and developing automatic pattern recognition and real time target defection/identification techniques.

FREDERICK W. RAHRIG
Project Engineer

Section 1

INTRODUCTION

1.1 PROGRAM OBJECTIVE

Under RADC Contract F30602-78-C-0017 PAR Corporation has provided engineering services to develop an Image Feature Manuscript Generation capability at the RADC Image Processing Facility. The capability developed here will serve as an experimental facility on which to evaluate advanced techniques of potential application to the Automatic Feature Extraction System (AFES). The AFES is being developed under RADC Contract F30602-78-C-0080.

This effort has been directed at providing an improved interactive means of classifying photographic imagery and of addressing and displaying graphics for the purpose of automatically categorizing image features, generating feature boundaries, and editing the shapes and locations of these boundaries.

At its outset this effort utilized the existing RADC image processing/pattern recognition software and hardware including the PDP-11/20, COMTAL Display, and the Xynetics plotter. While the general goals remained the same, the implementation strategy changed near the one-third schedule date. That is, plans to upgrade the PDP-11/20 to a PDP-11/70 system were affirmed and this led to the decision to make the Manuscript Generation system compatible with the AFES which also runs on a PDP-11/70 under the UNIX operating system.

Deliverable items under this contract have included computer software and descriptions of computer programs developed under this effort. An

interesting, and useful, feature of this system is that software documentation is available on line to the user.

Under this contract an atmospheric modeling algorithm developed at Air Force Avionics Laboratory (AFAL) was integrated into the system software structure.

1.2 BACKGROUND

The Digital Land Mass Simulation (DLMS) Data Base provides parametric input for radar scene simulators such as the Experimental Radar Prediction Device (ERPD). It is anticipated that this data base may also prove efficient for simulating displays for other types of sensors (e.g., Infrared) as well. Accurate simulation requires that the appropriate information be extracted from the source photography and entered without error into the DLMS data base. Current operational techniques for encoding relevant information is a time-consuming process including the manual tracing of terrain category boundaries.

As an enhancement of the present feature encoding process, it is desirable to automate, using digital image processing technology, procedures for generating data files to support mapping, charting, and geodesy products from analog and digital source imputs. The software system developed under this contract will serve as a test and evaluation facility for the development of advanced techniques of potential application to this problem.

Prior to this contract, RADC has been responsible for the development of image processing and pattern recognition technology which is directly applicable to the image feature extraction problem. The algorithms and software have been developed and applied under several contracts utilizing the

PDP-11/20 Image Processing System at RADC. This system was a display-oriented, mini-computer system dedicated to developing, testing, and evaluating computer techniques for information extraction from and interpretive exploitation of digital imagery according to the needs of the intelligence and reconnaissance activities of the Department of Defense. The software system was highly interactive and extremely flexible as has been demonstrated through the application to problems concerning a variety of image types including the following:

- black-and-white and multispectral reconnaissance photography
- side-looking synthetic aperture radar imagery
- forward-looking infrared imagery (FLIR)
- ERTS imagery

The objectives pursued have ranged from terrain categorization to tactical target detection and identification, and to image preprocessing for the purposes of visual image enhancement.

Some of the relevant developments at RADC prior to this effort include the following:

DICIFER - RADC Image Processing System which provides an interactive image processing capability for research related to feature extraction and classification from reconnaissance imagery.

OLPARS - On-Line Pattern Analysis and Recognition System for the interactive processing of multivariate data in a vector format.

Terrain Classification - Automatic extraction of terrain categories to be used in generating reference maps for radiometric sensors. Both black-and-white and color source imagery were utilized for efforts in support of Air Force Armament Test laboratory requirements.

ADET - (Advanced Digital Exploitation Techniques) included the design and implementation in software of techniques to compensate for radiometric and geometric anomalies encountered during the image digitization process. These techniques were tested on images scanned with the RADC Computer Eye. A sensor/sun position and atmospheric corrections program was also developed. The atmospheric modelling program developed operated as a user program on the RADC Image Processing System.

The collection of the above developments served as a significant base on which to develop an interactive image feature manuscript generation capability on the RADC Image Processing Facility.

The system developed under this effort is described in the next two sections. Additionally, two items of special interest are described in Section 4 (Region Growing) and Section 5 (Atmospheric Model).

Section 2

SUMMARY OF APPROACH

2.1 OVERVIEW

The approach taken has been to implement an image manuscript generation capability in software for the RADC facility. The basic software which has been implemented may be characterized under the following subdivisions:

I. Image Input

II. Preprocessing

III. Measurement Extraction

IV. Classification

V. Post Processing

VI. Image to Graphics Conversion

VII. Graphics Output

2.1.1 Image Input

Image input may be from magnetic tape or scanner. A calibration step wedge must be available for the image if the atmospheric modeling software is to be used.

2.1.2 Preprocessing

Preprocessing functions are of two types. The first involves adjustments to image gray levels to accommodate changes due to time, sensor position, or

different environment factors. The current capability was the AFAL-developed atmospheric model. The second type is concerned with the enhancement of the digital image for the purpose of display and analysis. Included are means for contrast enhancement, edge conditioning, and image analysis (via histogram displays). Certain of these functions are available under cursor/trackball control at the display monitor.

2.1.3 Measurement Extraction

This process involves the selection of representative sample sets of surface material categories contained within the imagery. This is done interactively by the user.

For the chosen samples, the next step is the calculation of measurements which are characteristic of sample properties. Current measurements available include the gray value of the pixel and measurements of texture for the area surrounding that pixel. Texture measurements are modeled after certain of those previously used by Haralick, Mitchell, or Hsu.

2.1.4 Classification

Once measurement vectors have been generated at the previous step, a classification logic is generated as a function of measurement statistics. The intent here is to devise a classification logic which partitions the n-dimensional measurement space in such a way as to separate the various surface material categories used within the sample set.

After the logic is created, it can be applied to the entire image, the results theme-encoded and displayed for on-line analysis.

2.1.5 Post Processing

This module permits the operator to review the results of the classification process and to edit as necessary. Editing is performed through the use of the interactive trackball and the color monitor.

2.1.6 Image to Graphics Conversion

This module is concerned with converting the output of classification (i.e., theme-encoded image) to a graphics format. To do so the boundaries of regions of homogeneous material type are extracted and converted to a lineal vector format.

2.1.7 Graphics Output

Within this module the boundary data extracted above is converted to a format compatible with the RADC Xynetics plotter. A nine-track magnetic tape is generated for transfer to the latter device.

2.2 PDP-11/20 SOFTWARE DEVELOPMENT

During the course of this effort, several major software developments were performed in assembly language for the PDP-11/20 based image processing system at RADC. The first of these involved the development of a stand-alone "region grower" capability under DOS (Disk Operating System) using the COMTAL image display for interactive seed point selection. This initial system allowed test and evaluation of the region grower algorithms and refine same to the current configuration. This system contained all the necessary

functions for display, operator interaction, and processing to facilitate such testing. This work was discussed in detail in two previous reports to RADC as follows:

1. Technical Memorandum: "Interim Report on Region Growing", by David H. Taenzer, PAR Report Number 78-14, 25 April 1978.

2. "DEMO User's Manual", by David H. Taenzer, PAR Report Number 78-18, 18 May 1978.

The DEMO system has been used extensively by both RADC and PAR personnel and any difficulties encountered have been resolved and retested. Results of such experimentation have contributed to the design of the current region growing capability.

Other major tasks performed on the PDP-11/20 system were the following:

1. To integrate the region grower algorithms into the framework of the DICIFER system. This involved mapping the appropriate software to run within the DICIFER executive and to be accessible via system option selections.

2. To fully integrate the COMTAL display subsystem into DICIFER as a replacement for the Spatial Data Systems display. As such the COMTAL display became available for the region grower capability within DICIFER and numerous other functions which utilized image displays could also utilize the COMTAL device. This represented a major software development task because of the numerous functions which utilize the display and due to the fact that DICIFER consists of a custom built operating system coded in assembly language.

3. To develop a dynamic transfer function capability on DICIFER. In particular, the COMTAL display and trackball were used interactively to achieve the desired enhanced image. That is, by controlling the trackball the operator could modify the transfer function and immediately see the results on the image display. An option was also provided whereby the user could elect to save (create a new file on the RP02 disk) the results available at a given state of the function table.

2.3 PDP-11/70 SOFTWARE DEVELOPMENT

The Image Feature Manuscript Generation (IMFG) effort has involved experimentation with and development of new techniques applicable to feature extraction (the region growing capability is an example of this). The original intent was to utilize the body of software available under DICIFER and to develop the "Manuscript Generation" system around this to run on the PDP-11/20 hardware configuration. Two events took place by summer 1978, however, to affect this approach. That is, there existed a commonality in concept between software applications modules to be provided under the Image Feature Manuscript Generation effort and those provided within the AFES (Automatic Feature Extraction System). AFES is based on a PDP-11/70 configuration and is being developed under RADC Contract F30602-78-C-0080. Also, it was anticipated that the PDP-11/20 hardware which comprised the RADC Image Processing Facility would be upgraded to a PDP-11/70 system in late 1979.

Based on these assumptions, it was determined to be advantageous to develop the Manuscript Generation system to run under the UNIX operating

system so that applications modules software common to that system and the AFES need not be developed twice. Moreover, this approach provided that the resulting systems would be compatible, thereby facilitating long-term experimentation and development efforts. For example, AFES systems delivered to DMA may be used to test and evaluate AFES capabilities via PDO (Pilot Digital Operations) experimentation while the system to reside at RADC may be used to investigate and design new and more powerful techniques for automatic feature extraction. As new techniques are developed, they may then be easily installed on the DMA versions of the AFES.

Another benefit inherent in this approach is that the total system to be installed on the RADC PDP-11/70 contains a comprehensive set of image processing functions which are generally applicable to a variety of problem areas in addition to manuscript generation. These include image enhancement, target detection/recognition, and terrain classification and are applicable to various sensor data including synthetic aperture radar (SAR), infrared (video or line-scan format), or electro-optical systems.

In order to implement the approach reference above, the plan listed below has been followed. As such this describes the development of the IMAGE FEATURE MANUSCRIPT GENERATION capability under Contract F30602-78-C-0017 using, in part, developments made within the AFES system under Contract F30602-78-C-0080.

<u>AUTOMATIC CLUSTERING (Region Growing) -</u> Developed under -0017 on PDP-11/20. Added under UNIX system under -0017 and extended to allow large windows (full image) by exploiting the display memory configuration of the AFES.

TEXTURE MEASUREMENTS - Analyzed under -0017 and added to UNIX system under -0017. Expanded capabilities to be added under -0080.

OPERATING SYSTEM AND EXECUTIVE - Software system design and specifications, operating system considerations, and task management software have been primarily addressed under -0080. This uses UNIX software system and enhancements thereof and follows doctrine established under -0080.

IMAGE INPUT - General handling of image and data files is being addressed under -0080.

PREPROCESSING - AFAL-developed atmospheric corrections program will be integrated under -0017. Preprocessing functions for image enhancement are being addressed under -0080. Image histograms, convolution filters, and median filtering have been achieved under -0017.

AUTOMATIC FEATURE EXTRACTION AND CLASSIFICATION - Analysis and design being performed under -0017. Software primarily being developed under UNIX through -0080.

POSTPROCESSING (Edge Extraction) - Partially developed under -0017 via classified image boundary linearization. Implemented under UNIX through -0017. Modified capability for final system to be developed under -0080.

INTERACTIVE GRAPHICS - Generalized software being developed under -0080. This module contains editing of classified results and conversion to output formats.

GENERAL COMMENTARY - Certain AFES functions will take advantage of the RAM memory available within the DeAnza display system. The advantages available

in this way to the AFES are not immediately available to the MANUSCRIPT GENERATION system due to a different display system at this time. In general, whenever the display system is at issue within an algorithm, some modification will be necessary to go from AFES to MANUSCRIPT GENERATION.

Whereas the total system under development supports a great diversity of approaches toward image processing experimentation, a particular approach is cited here to provide an overview of system utilization. That is, an approach for experimenting with image feature extraction is cited by briefly describing a logical sequence of functional steps within such a process.

Step 1: Input digitized image data into the process. The data may be images scanned by the LIPS at RADC, from the on-line scanner on AFES, or on magnetic tape from other sources.

Step 2: The experimentor may now review and analyze the digitized image using interactive display capabilities. Image enhancement and filtering functions might also be applied to aid in the analysis process.

Step 3: At this point the operator may elect to apply adjustments for atmospheric conditions. To do so, the user executes an atmospheric modeling algorithm developed by, personnel at AFAL and interfaced into the UNIX system by PAR.

Step 4: Automatic region growing may be performed on the image. This capability segments the digitized image into autonomous parts according to tone and texture properties. The operator can view the results on the display as are being calculated and vary input parameters to modify the process.

Step 5: At this point it may be appropriate for the user to experiment with image classification. Typically this will involve displaying the digitized image and interactively selecting (via the cursor) samples of various image features (classes). The user then specifies the

list of measurements to be used. These can be selected from available tone and texture measurements. The image is then classified according to a user specified logic and a theme coded image is output to the display. At this point the user may elect to edit the results by interacting at the display and using the cursor and keyboard to input the desired changes.

Step 6: Once the user is satisfied with the results at Step 5, he may elect to encode the data into a manuscript format instead of the raster format used up to this point. In particular, contiguous regions of like pixels (i.e., of the same class) can be grouped together to form a feature. The feature is represented by the boundary which encompasses it and by its category. Results of this process can be reviewed visually by displaying the boundaries as overlays in the on-line monitor or by creating a manuscript tape. The latter may be transported to the Xynetics plotter and used to plot feature boundaries. Additional work related to this area is being done within the AFES to provide a general basis for associating data on categorized pixels to achieve higher level classification regarding image features. This relational data base concept is currently under design and development within the AFES system.

2.4 OVERVIEW OF THE AFES

The burden placed on the human operator in an intense image exploitation environment is a heavy one indeed. As a means to minimize this burden and therefore increase throughput rates and accuracy, automated object detection and identification are required. PAR has extensive related experience in these areas. An especially noteworthy example is that of the AFES (Automatic Feature Extraction System), which is currently being developed by PAR under RADC Contract F30602-78-C-0080.

The primary purpose of AFES is to validate computer-assisted and semi-automatic feature extraction techniques as a means for producing digital maps, and to provide the basis for design and specification of an actual production system. By design, the AFES also represents a powerful research tool with applications to a variety of automatic image processing problems.

Development goals for AFES can best be understood when viewed in the context of a generalized approach to pattern recognition. Such a generalized approach is composed of three research paradigms: decision theoretic, or statistical pattern recognition; syntactic pattern recognition; and artificial intelligence. Of these, the decision theoretic approach has been implemented in AFES, and the syntactic and artificial intelligence approaches have been proposed as additions to AFES. The decision theoretic approach is applicable to classification of individual pixels based on feature vectors associated with pixels or their immediate neighborhoods, and has been successful in identifying local properties.

AFES is designed to be a multi-user facility that is easily modifiable, modular, and remotable. Multi-user implies that expensive resources can be

shared among users. Easily modifiable means that new algorithms can be easily developed and incorporated into the system, new algorithms can easily use all techiques and algorithms which have been previously implemented on the system, and new processes can be easily structured from a variety of algorithms and techniques. The modular characteristic supports easy modification by making previously programmed techniques available to new algorithms, allow processes to be reconfigured from various modules to permit execution on various processors within the system, and allows clean system design, implementation, maintenance, and modification. Remotable implies that users of AFES can be located at long distances from the central hardware.

There are a number of AFES functions that are specific to the production of digital maps that should be mentioned. A precision scanning system is provided for image input from photographs, maps and charts. This includes automatic registration via warping transformations so that, for instance, photographic data can be overlaid with map data. The warping transformations rely on identification of control points in the two images to be registered, and are computed automatically given that information.

The scanning and display systems are designed to permit viewing of stereo input photographs and stereo displays of output images.

Additionally, the scanning system permits rapid changes in field of view via a zoom system, so that subpictures within the input image can be scanned at higher resolution, while maintaining correct registration with other input sources.

Three display systems are provided in the AFES system. Two 1024 x 1024 monochrome displays are used to display stereo imagery, to display a picture

and a subpicture simultaneously, to display a photographic scene and corresponding map section, to display images with various overlays, or to display two conjugate images in stereo.*

A 512 x 512 color display may be used to display true-color input data, or to generate pseudo-color displays for use in, for example, evaluation of classification procedures.*

A preprocessing module is provided to enhance either all or selected parts of an image. Algorithms available include dynamic range changing, equal probability quantization, intensity thresholding, gray scale histogram computation, and edge detection.

Finally, editing and data recording modules are provided for recording feature data collected to construct the digital map of the input scene.

Software modularity in AFES is achieved by using the UNIX operating system, by having a single, unified file system that supports all types of data and procedure files, and by development of the "AFES discipline," a set of programming practices and conventions common to the entire system.

The UNIX operating system is specifically designed to support modularity, ease of development, and multiple users. The Programmers Work Bench UNIX provides extensive on-line editing and program development facilities, plus provisions for automatically cataloging different versions of AFES systems. The goal of providing to the user a flexible and easy-to-use image processing test bed facility implies easy interface of user-written routines to the AFES

* Whereas the AFES uses DeAnza display system, the IFMG is currently operating with a COMTAL (512 x 512) color display system. It is anticipated that DeAnza systems will be added to the IFMG in the future.

system. This is accomplished through an interactive program that generates the necessary software interfaces for a user subroutine, reducing to an absolute minimum the knowledge the user must possess about the AFES system. In addition, a library of software support routines will be available to the user to permit generation or decoding of AFES data files. Other support routines are provided to permit execution of other processes by the user program, and to control interaction of the user with experimental programs.

The overall effect of these software support tools is to allow a new experimenter to begin using the AFES system with a minimum of knowledge about the system itself. As the user's familiarity with AFES grows, he or she will be able to develop more sophisticated and powerful programs, but this familiarity will not be a prerequisite for system use. Further details on the AFES are provided in Section 3.

Section 3

SYSTEM DESCRIPTION

3.1 INTRODUCTION

As discussed in Section 2 the development of the IFMG has been in parallel to the development of the AFES. The latter effort has been much broader in scope, however. The key difference between the systems at this time is that the IFMG does not possess the full complement of displays that are resident within the AFES. It is anticipated, however, that these displays will be added to IFMG in the future. The remainder of this section details the attributes of the AFES (and, therefore, of the IFMG).

3.1.1 AFES Purpose

AFES has been designed primarily as a test-bed facility for image processing and exploitation by machine. Accordingly, strong emphasis has been placed throughout its development on such things as modularity, user interfaces, and software support. These goals will be discussed in more detail in the next section.

The original AFES development effort was directed toward development of a prototype system for production of digital maps. Although the development goals were redefined early in the project to fulfill the need for a test-bed system, the importance of the map-making application has not been neglected. Accordingly, a secondary purpose of the AFES is to support development of digital interactive image processing and feature extraction capabilities for generating data files to support mapping, charting, and geodesy products from analog and digital source inputs.

3.1.2 AFES Goals

The AFES design specifications require that it be a multi-user system which is easily modifiable, modular, and independent of image source.

3.1.2.1 Multi-user Facility

Multi-user implies that expensive resources can be fully utilized by sharing among users. This has been accomplished by use of a work station configuration, in which each user has, for his exclusive use, interactive devices and minimal computational capabilities appropriate to his task. Resources shared with other users, including mass storage devices, special types of processors, and image input devices, are controlled by a large central processor which is linked to a number of work stations.

3.1.2.2 Easily Modifiable

Easily modifiable means that:

- New algorithms can be easily developed and easily incorporated into any part of the system.

- New algorithms can easily use all techniques and algorithms which have been previously implemented on the system.

- New processes can be easily structured from a variety of algorithms and techniques.

3.1.2.3 Modularity

Modularity implies that:

- Previously programmed techniques are available to new algorithms.
- Processes may be reconfigured from various modules so that these processes can execute on various processors available within the system. Different flavors of processing can be easily developed.
- System design, implementation, maintenance and modification may be clean and efficient.

3.1.2.4 Independence From Image Source

AFES must be independent from its image source in order to retain compatibility with all present and future image sources which it may be used to exploit. While each new image source may require a different hardware device to digitize the image data and different software modules for image queueing and formatting, the result of the input process will be images in standard AFES file format. The AFES file format has been designed for maximum versatility, and will accommodate both single channel and multichannel imagery. Eventual compatibility with all digital imagery is an AFES goal.

3.2 OVERALL AFES STRUCTURE

This section will present more detailed information about the AFES workstation configuration, software system, executive control, program development aids, and applications software.

3.2.1 The Workstation Configuration

As mentioned in the previous section, the workstation concept provides for a set of dedicated interactive devices for each user. The type of workstation used depends on the operator's task. In general a number of workstations will be linked to one master processor which allocates shared resources among users.

3.2.1.1 Master Processor

The master processor is a large minicomputer (PDP-11/70) with a variety of image input, mass storage, and processing resources (Figure 3-1). Input images may be provided on magnetic tape, and a number of tape drives are provided for access and copying of image data. The design includes a scanner system which is linked to the processor via a dual-ported disk system so that film, map, or chart data may be digitized and stored on the disk and utilized by the system as needed. A second large capacity disk system stores source images and intermediate results of image processing functions executed on the master processor. Processing resources include, in addition to the capabilities of the PDP-11/70, a floating point array processor which is used to perform certain types of tasks involving numerical computation on large blocks of data.

3.2.1.2 Work Station

A hierarchy of work stations is provided which seeks to match the hardware configuration used with the task to be performed. The types of workstations have been termed the Program Development Station (PDS), the Technique Analysis and Development Station (TADS), and the Full Function

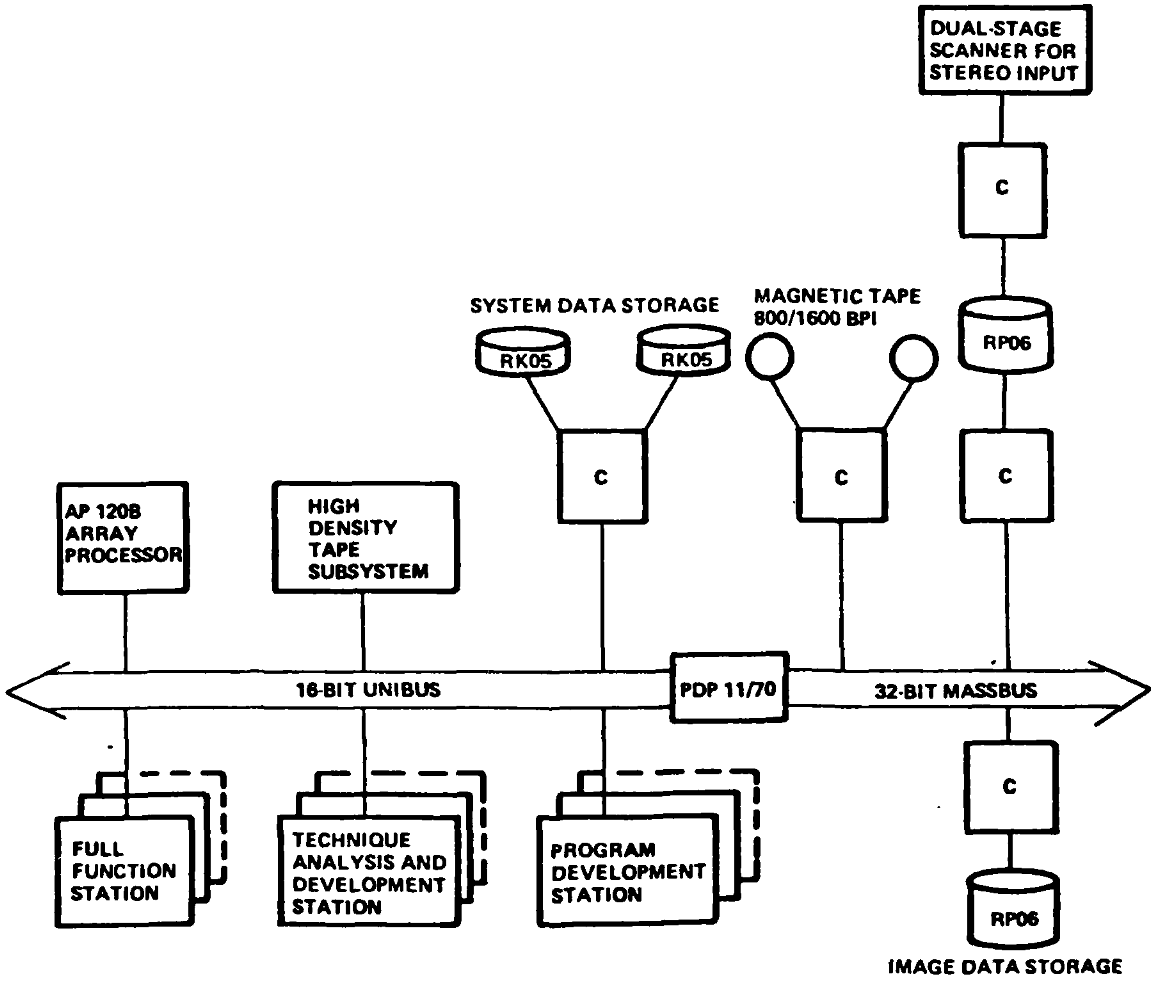


Figure 3-1 Master Processor Configuration for AFES

Station (FFS).

3.2.1.2.1 Full Function Station

The FFS (Figure 3-2) provides the full complement of image processing and interaction capabilities. A color display system is included on which the user may view source imagery or the results of processing operations. Two high resolution monochrome display systems and a stereo viewer are provided to allow display of stereo imagery. Each display system has function buttons, a trackball, hardware cursors, and overlay memory to accommodate operator interaction and display of auxiliary data. The FFS configuration provides the environment necessary for integrated testing of image processing functions and design and implementation of the types of software systems envisioned for production of digital maps. The display system is controlled by a PDP-11/34 display processor which also provides a minimal processing capability. In particular, operations which require frequent and/or random access to image data, but do not perform complex computations are well suited to execution on the display processor. These may include such things as histogram computation, contrast modification, edge detection, simple geometric transformations, and other preprocessing or enhancement operations. Image data may be transferred to and from the master processor via a high speed parallel data link.

Facilities for operator interaction for the FFS are designed to minimize the knowledge required to use the system. Commands issued by the workstation user may refer to processes which are executed on the master processor or the display processor. To simplify operation incoming commands are automatically

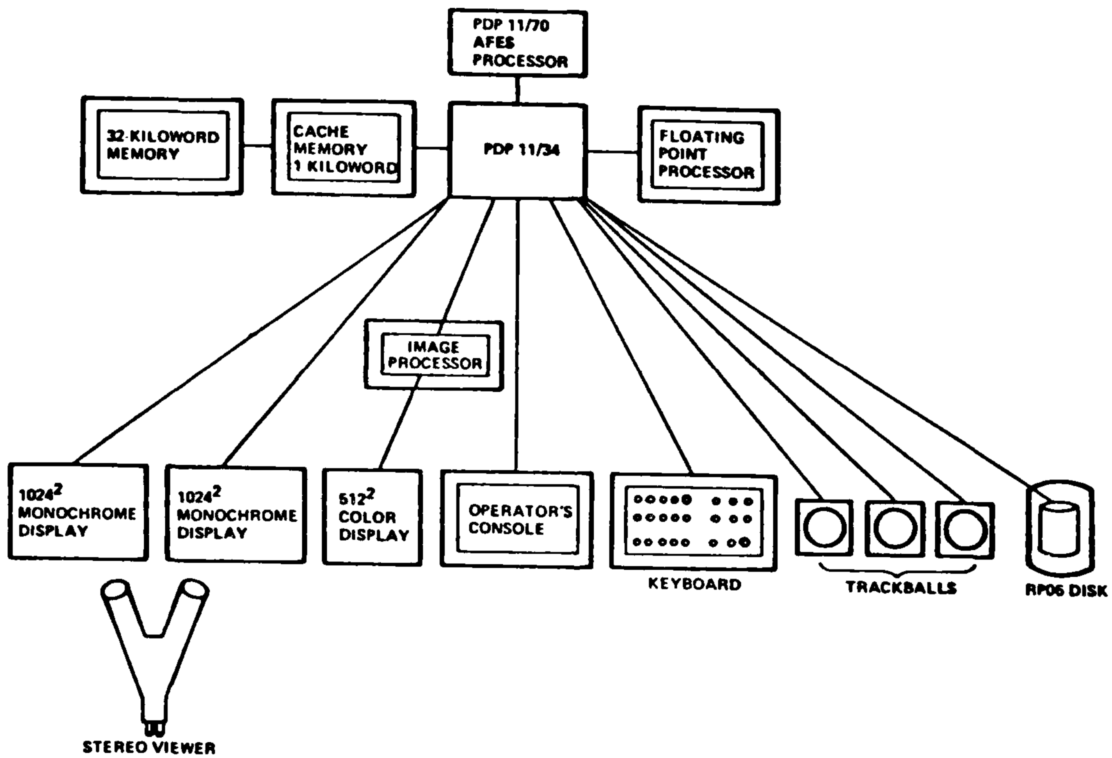


Figure 3-2 Full-Function Station Configuration as Exists in AFES

sorted by the display processor's command interpretor. Those which run on the display processor are executed immediately, while others are transferred to the master processor's command interpretor.

In general, programs which require operator interaction are executed on the display processor. In some cases a single command may start a process on the display processor which will interact with the user to obtain input data or parameters, then start a "batch" type process on the master processor to perform a computation using the user's input data. Most user interaction occurs via trackballs, cursors, and pushbuttons.

An additional interactive device is a voice recognizer, which may be trained by each operator to interpret simple vocal commands and issue the appropriate character strings to the display processor. This allows the operator to enter commands when both hands are occupied controlling trackballs.

3.2.1.2.2 Technique Analysis and Development Station

The TADS provides the same hardware configuration and functional capabilities as the FFS with the exception of the dual monochrome displays and the stereo viewer. It is envisioned as a test station on which new image processing algorithms can be developed and tested, with results analyzed with the color display system. Since full production capabilities are not required, viewing of stereo imagery will in general not be necessary, although the color display may be used for anaglyph stereo display if needed.

3.2.1.2.3 Program Development Station

The PDS consists simply of a CRT terminal which is linked to the master processor. It is designed for the user who simply wishes to edit and compile programs, and to execute programs on the master processor for which image display output is not needed. The multi-user operating system used for the master processor can accommodate a large number of these minimum-configuration stations without noticeable degradation in response time.

3.2.2 AFES System Software

The AFES system software has been designed to support the features described in Section 3. The UNIX operating system has been used to provide multi-user time sharing capability. UNIX provides a convenient file structure which supports the feature of modularity and independence from the image source. A unified system of programming practices, including software control and systematic documentation, has been implemented to make AFES easily modifiable.

3.2.2.1 UNIX Operating System

The UNIX operating system was designed to support modularity, ease of developing, and multiple users. The "Programmer's Workbench" (PWB) version of UNIX has been used for AFES. PWB/UNIX provides the following features particularly important in its application to AFES:

- A hierarchical file system.

- A flexible, easy-to-use command language.

- Ability to execute sequential, asynchronous, and background processes.

- A powerful context editor.

- Flexible document preparation and text processing systems.

- A high-level programming language conducive to structured programming (C).

- The programming languages BASIC and FORTRAN.

A number of these features will be described in more detail in following subsections. The UNIX file system is very important to AFES, as it contributes significantly to software modularity and image source independence. The UNIX command language, called the Shell, is used to implement the AFES "Image Processing Language", which controls file access and user processes and greatly simplifies execution of image processing functions. The capability for background processes provides for a smoother process flow in execution of statistical pattern recognition routines on images, since some processes, such as classification, can be run in the background while the operator is using the terminal for other routines.

3.2.2.2 AFES File System

The AFES file system is based on the standard UNIX file system, and many AFES features are achieved through careful organization and implementation of file structures. The UNIX file system will be described first, followed by its use in implementation of AFES program and data files.

3.2.2.2.1 UNIX File System

The PWB/UNIX file system consists of a highly uniform set of directories and files arranged in a hierarchical tree structure. Each node in the tree is either a file or a directory; if it is a directory it will have branches to lower level nodes. Basic features of the file system are:

- Simple, consistent naming conventions. Names may be absolute or relative to any directory in the tree.
- Mountable and de-mountable file systems and volumes.
- File linking across directories.
- Automatic file space allocation and de-allocation transparent to the user.
- Flexible file and directory protection modes, with read, write, and execute access controlled for individual owners, a group of users, or for all users.
- Facilities for creating, accessing, moving, and processing files, directories, or sets of these in a simple, uniform way.
- Treatment of each physical I/O device, ranging from interactive terminals to main memory, as a file, allowing uniform file and device I/O.

3.2.2.2.2 AFES File System

Description of AFES files requires a slightly more detailed description of the UNIX file structure. If one considers a node in the directory tree to

be a directory called "dname", then entries in this directory are referred to by a "pathname", which for the entries in dname would be "dname/name1", "dname/name2", etc. The UNIX file system has as its root node a directory containing names of a large number of other directories. Entries in the AFES root directory include, among others, "/i", which is a directory leading to image files, "/u", which branches down to personal user files, "/usr", which contains system routines, and "/tmp", which contains temporary files. The three root node directories most important to AFES are the /i, /u, and /w directories. The /i and /u directories will be described in sections which follow. /w is a working directory for temporary storage of image files during processing sessions.

File Ownership. Both image and user files are uniquely associated with a particular user. Each file is coded as to access privileges, with the code indicating read, write, and access privileges to the file owner, a specified group of users, or to all users. This access control provides file protection and assumes an important role in the UNIX software control system.

Image File Directory. The /i directory leads to a tree of image files. Each image has assigned to it a directory with at least two entries, a header file called "hdr", and a "data" file. The header contains information about image size and format, plus a list of parameters which relate the image to geographic or earth coordinates. Any process which is executed using the image as input leads to another directory entry which is just the name of the process. This "process" entry is itself a directory containing header and data files for the output image produced by the process.

The sequence of directory nodes in /i is designed to be a highly organized record of all processes which have been executed on an image. This directory structure may be reviewed with the "examine" command, which allows the operator to interactively examine the directory structure. Automatic restart capability is provided by storage of all status information in the file structure so that a particular processing environment can be automatically invoked when a user logs onto the system.

User File Directory. Each user has a personal directory containing programs under development. While this directory may contain only names of various files, it more often contains names of other directories which divide the user files into categories.

AFES Directory. The AFES utility and applications programs are all contained in a user directory which has the pathname "/u/afes". There are a large number of directories under /u/afes; they include the following:

cmd	program development commands
bin	AFES administrator commands
bin$z	command language (Shell) routines, release no. $z
system	system utilities such as control commands for the interprocessor link
lib$z	master processor library routines, release no. $z
obj$z	master processor executable modules, release no. $z
smlib$z	workstation (PDP-11/34) library routines, release no. $z
smobj$z	workstation (PDP-11/34) executable modules, release no. $z
papers	a collection of technical papers

Successive updates of the AFES software, incorporating new programs and changes to existing programs, give rise to new release numbers. "$z" is a Shell variable which contains the current release number. Hence "/u/afes/lib$z" is the pathname to master processor library routines for the latest AFES release. Command language routines, written in the Shell command language, are the top level commands which are executed by the user, and in

turn start up the executable programs. These commands constitute components of the AFES "Image Processing Language". Library routines are subroutines used by many programs, which are combined with the calling program to make an executable module. Object modules are compiled versions of main programs.

Separate library and object directories are provided for the master processor and the workstation processor. While the workstation routines are maintained by the same software control system as the master processor routines, they are compiled in a slightly different way due to differences in the capabilities of the processors.

3.2.2.3 AFES Programming Practices

The goals of modularity and easy modifiability are attained through careful structuring of programs, attention to consistent documentation, utilization of system I/O routines, standard image formats, and strict software control.

3.2.2.3.1 Program Structure

AFES applications programs are written in a way which allows maximum flexibility and versatility. Standard software interface routines are used whenever possible, such as the "automatic window" code, described in a later section. Subroutines are written to provide as much application independence as possible, so that they can be used by a large number of programs, thus minimizing the number of subroutines.

3.2.2.3.2 Consistent Documentation

Standard documentation formats are used for all programs. Each source code file has standard documentation "boilerplate", which identifies the author, describes the program, and tells how it is used. Each program also has a separate documentation file which describes the program at the level required by the typical user i.e., calling sequence, function, options, etc. Programs which constitute executable commands have an entry in the on-line AFES manual instead of a documentation file. A hard-copy off-line manual will be provided as well which contains all of the manual and documentation files.

3.2.2.3.3 System I/O Routines

UNIX provides standard input and output files which are used whenever possible. The user may specify any file or terminal to be used as standard input or output, or may transfer the output of one program directly to the input of another using these standard I/O facilities.

3.2.2.3.4 Standard Image Format

The increasing variety in types of image sensors available for acquisition of mapping information has lead to a concomitant increase in image formats. AFES achieves a good deal of its versatility by reducing image data from all types of sources to a single, standard image format. Thus the image processing operations are independent of image source, and can accommodate both current and future forms of imagery.

3.2.2.3.5 Software Control

Software control will be discussed in more detail in the next section; it basically consists of protection features which prevent unauthorized modification of AFES programs, assures that program changes are documented in a consistent fashion, and provides for automatic recompilation of programs using program interdependence records.

3.2.3 The AFES Executive

The AFES Executive refers to a collection of programs which control execution of processes, maintain software control, aid in program development, and provide interfaces to AFES documentation.

3.2.3.1 Process Control

The process control function of the executive consists of the set of command language routines which are executed by users and which in turn start up a variety of systems and applications programs. This capability greatly simplifies command string structure, since the command language routine can execute the proper sequence of executable modules base upon a simple set of flags and arguments provided by the user. It is this set of command language programs which make up the AFES Image Processing Language.

The UNIX Shell language allows the user to define a wide variety of variables which may be used to simplify command structure. For instance, a user wishing to use the paradigm support software for statistical pattern recognition may define a "working image" with which he wishes to experiment. The pathname of this image is saved as a variable which may be accessed by Shell routines which use the working image as input. Thus the user need not

specify a possibly long and complicated pathname each time he executes a command which operates on the working image.

3.2.3.2 Linkage to Paradigm Support Routines

User definition of a working image constitutes one feature of the software linkage which AFES provides to support particular types of pattern recognition paradigms. At present only statistical pattern recognition is supported; possible future extensions include support of syntactic pattern recognition and artificial intelligence paradigms. Paradigm support implies definition of particular variables, data file structures, and commands.

3.2.3.2.1 Variables

The pathname for the current working image constitutes one important variable as described in the previous subsection. Other variables defined include pathnames to directories containing executable modules and shell routines, and directories containing intermediate data files.

3.2.3.2.2 Data File Structures

File structures are set up to accommodate storage of intermediate and output data required by the particular pattern recognition paradigm in use. For statistical pattern recognition, for example, a highly ordered file structure is formed to contain data describing image measurements, training regions, and classifier statistics.

3.2.3.2.3 Commands

Specialized commands are defined. For statistical pattern recognition these include, for instance, "get region", which allows the operator to define

a training region, "train", which extracts measurements from the training region, and "classify", which executes the classifier on the entire image.

3.2.3.3 Software Control

AFES has an extensive source code control system which is maintained by the AFES Executive. This system utilizes two UNIX components, the Source Code Control System (SCCS) and the "Make" command, to assure system integrity.

3.2.3.3.1 SCCS

SCCS maintains a record of all changes which have been made to a program's source code, making it possible to reconstruct any earlier version of a program at any time. Each time a user modifies one of his programs, he is required to provide a short description of the reason for the change. This generates a historical record of a program's evolution. The AFES user directory (/u/afes) contains source code for each program which is part of the AFES system, thus preventing proliferation of multiple copies of a program which may or may not be identical.

3.2.3.3.2 Make Command

The UNIX "Make" command utilizes file interdependency data in performing recompilation of programs which have been modified. When a program is placed under AFES Executive control, the author specifies the names of all subroutines or other files which the program utilizes. This information is recorded in a "Makefile" placed in the same directory as the program. When a file is modified the AFES Executive can poll all of the Makefiles to find and

recompile all programs which depend on the file which has been modified. Usually the recompilation is performed, giving rise to a new "release" of AFES, after a number of programs have been changed.

3.2.3.4 Program Development

A number of program development aids have been provided to simplify the task of writing a new program or modifying an existing program. These aids allow the user to easily add new programs to the AFES software control system, and prompt the user to provide the information necessary to create entries in the proper Makefile. Commands are provided which allow a user to access an existing AFES program, edit it, and record the changes in the appropriate SCCS file.

3.2.3.5 Documentation Access

AFES provides several standard commands which allow the user to access on-line documentation for programs under AFES control. Every AFES command has a short usage information file accessed by executing the command "help <command name>". This command prints out (on the user's CRT) the proper argument sequence for the command, indicating which arguments are optional and which are required.

An on-line AFES manual is maintained, and may be accessed by typing "man <command name> afes". This provides extensive documentation about the program, including argument list, functional description, and files used.

Every file that is not an AFES command, including main programs, subroutines, and include files, has associated with it a documentation file

which may be obtained by typing "doc <file name>". The resulting output describes the program, author's name, calling sequence, etc.

The bottom level documentation is provided with the program source listing. In addition to frequent comments interspersed with the on-line code, a standard documentation section is provided at the beginning of the program, listing the author's name, the files, subroutines, and macros used, a program description, the compile string, etc.

3.2.4 User Programming Support

A number of AFES features are provided to simplify production of user programs. These features include standard documentation format, software interfaces to the AFES file system, and various utility subroutines.

3.2.4.1 Documentation

The AFES documentation format was described briefly in a previous section; program development aids are provided which allow the user to produce the necessary documentation for his program with a minimum of time and effort spent. The documentation commands "doc" and "man" operate on text files by invoking the UNIX text processing function "nroff". This function performs extensive text formatting operations including automatic numbering of subsections, printing of headings, indentation, etc. AFES provides commands which give the user a standard format for production of the nroff source files, so that the documentation can be written by simply "filling in the blanks" in a prestructured document outline.

The source code documentation is written using the same type of prestructured "boilerplate" used for man and doc files. The programmer simply

fills in all of the information required, including his name, files used, program description, etc. Only the on-line comments provided with the source code are left up to the programmer's personal style.

Information for the "help" command is acquired by prompting the user for a one line synopsis of the command's calling sequence when the program is first entered under AFES control.

The result of the documentation support is that all AFES documentation is produced in a consistent format, and the ease of documentation encourages the programmer to produce the documentation concurrent with his development of the program. This helps to avoid the last-minute large-scale documentation efforts which so often plague delivery of large software systems.

3.2.5 Software Interfaces

One of the most common types of user programs anticipated in use of AFES as a test-bed system is the measurement extraction routine which uses as its input the intensity of a single pixel, or perhaps the intensities of pixels contained within a small window surrounding a single pixel. The user is usually concerned largely with the routine which operates on the pixels within the window, and would rather not have to worry about the mechanics of moving the window throughout the image. Routines of this type may perform, for example, smoothing, edge enhancement, or texture description operations. To meet this need AFES provides a number of include files, referred to collectively as the "automatic window code", which may be inserted in the user window processing code. This automatic window code interprets the program parameter string to obtain names of input and output files, sets up memory allocation for input and output image data, sets up the line-by-line and

point-by-point loops which move the window through the image, and performs all necessary data conversions to input and output data. This reduces the time required to produce this type of image measurement functions from a matter of days or even weeks to a matter of hours.

3.2.5.1 Utility Subroutines

The AFES subroutine libraries include a large number of functions which, although they are more likely to be used by more advanced programmers, save considerable programming time and help eliminate duplication of effort. These routines have applicability in the areas of file handling, display interaction, error handling, and numerical operations, and include the following:

file access	locate and open image header and data files
data conversion	convert output data from a given type to an arbitrary format as specified in output image header
input and output	perform I/O for single lines of image data
matrix manipulations	perform matrix addition, multiplication, and inversion
display routines	initialization of DeAnza display registers; cursor and trackball interaction
error handling	standard routines for printing error messages.

3.2.6 Applications Software

Three general classes of applications software have been provided with the current release of AFES. These include functions for experimentation with statistical pattern recognition, image registration and resampling for viewing stereo imagery, and general software for image display.

3.2.6.1 Statistical Pattern Recognition

The paradigm support software described in Section 3.2.3.2 plays an important role in statistical pattern recognition experiments, since it greatly simplifies execution of programs by maintaining records of the working image and previous processing executed on the image. Statistical pattern recognition is a "table driven" process in AFES. This means that the user constructs a table listing the measurements he wishes to extract from the image plus the name of the classifier he will use to process the measurement vectors. The table is accessed by commands such as "train" and "classify" to determine what particular measurement extraction and classification programs are to be used.

Future AFES enhancements will extend this concept into a complete image processing language, in which the operator can specify a list of operations he wishes to perform on the data (e.g. image enhancements, measurements, and classification). It will be possible to set up a sequence of operations on one image, i.e., a "do loop", changing parameters with each iteration, or to perform one sequence of operations on a set of images.

Support software for statistical pattern recognition includes measurement extraction programs which operate on pixels or pixel windows and produce

parameters used to classify pixels, interactive display routines which allow the operator to define training regions and display classification results, and a classifier which may be used to classify an image into various categories. These will be briefly described in the following subsections.

3.2.6.1.1 Measurement Extraction

Although the initial thrust of AFES development has concentrated on process control, program development, and paradigm support software, a small set of measurement extraction programs have been provided with the initial delivery. These include the pixel image intensity, smoothed versions of intensity obtained by a moving window average, a Laplacian edge enhancement measurement, and a Hsu contrast measurement, which is a texture measure. These routines can be cascaded to provide additional measurements, such as a smoothed version of the Hsu measurement.

3.2.6.1.2 Display Routines

Several interactive display routines have been provided to support statistical pattern recognition. One allows the operator to outline a training region on the display screen using a trackball to "draw" the boundary of the region. Multiple training regions can be defined for each output category, and the support software automatically records a description of the training region for use in accessing pixel data to train the classifier.

After the image has been classified another routine displays the original image on the color display, and uses the classifier output to give each pixel a color tint corresponding to its category. An interactive routine allows the operator to choose the color which will be associated with each output class.

3.2.6.1.3 Classifiers

The classifier which has been provided with the first AFES delivery chooses the output class of a pixel based on minimum Mahalanobian distance. This classifier uses a statistics file generated from the training set which comprises a statistical description of each output class. An unknown pixel is classified based upon the minimum "distance" to a training class.

3.2.6.2 Stereo Imagery

Software for support of stereo imagery includes image display, image warping, and control point registration routines.

3.2.6.2.1 Display

Image display routines allow transfer of image data from files on the master processor to any one of five image channels supported by the display system. These channels include the red, blue, and green channels of the color display and the channels associated with the two monochrone displays.

3.2.6.2.2 Warping

Image warping is provided via the affine transformation and either nearest neighbor or bilinear interpolation resampling. These allow one to transform an arbitrary parallelogram on one display channel to an upright full-size rectangular region on another channel. The transformation can perform rotation, translation, and scale change.

3.2.6.2.3 Pass Point Registration

An interactive display routine is provided which allows the operator to use the trackballs and cursors to define pass points in conjugate images on the left and right monochrome displays. The pass point coordinates are then used to compute a transformation which may be used to bring the left and right conjugates into proper registration for stereo viewing. The resulting stereo pair may be viewed in anaglyph on the color monitor, or on the monochrome monitors using the stereo viewer.

3.2.6.3 Display Software

General purpose software for the display system may be divided into three categories: display utility programs, image manipulation programs, and operator interaction programs. Many of these are provided in the form of subroutines which may be linked to user programs.

3.2.6.3.1 Utilities

Utility commands are provided which clear the image channels, clear the overlay channels, and initialize the display registers. A utility subroutine package is provided for use in setting up the memory mapping which allows access to the image refresh memory via direct addressing of the PDP-11/34 memory space.

3.2.6.3.2 Image Manipulation

These routines allow display of master processor image files on any one of the display channels and copying of image data from one display channel to another. Other programs allow one to view a full-size image with a zoom power

of 2, 4, or 8; a scrolling function is provided so that the zoom window may be moved throughout the full-size image via trackball control.

3.2.6.3.3 Operating Interaction

A number of subroutines are provided which allow control of image cursors with the trackballs and retrieve the coordinates of the cursors on exit. Another routine allows one to draw lines on the display screen to outline boundaries of regions.

SECTION 4

REGION GROWING

4.1 DESIRABLE FEATURES OF REGION GROWING

Image segmentation is the process of dividing an image into meaningful areas, or regions. A region consists of a spatially connected subset of picture elements (pixels). For example, in an aerial photograph, each region may correspond to some surface material category. In a picture of a person's head, the regions may correspond to parts of a head, such as the eyes, nose, mouth or ears. The basic idea of segmentation is to describe an image containing a large number of pixels with a substantially smaller number of regions in such a way that the significant picture information is retained. The definition of "meaningful areas" and "significant information" depends on what we are trying to do with the image. If we are trying to locate objects, then the segmentation should be done in such a way as to preclude having two different objects in one region. If we are trying to classify the pixels into various kinds of materials, then each region should contain only one class.

Since image segmentation is a type of data reduction, the next stage in the processing of the image will presumably operate on regions, instead of individual pixels. This is why most segmentation schemes are designed to lean in the direction of over-cutting, or creating extra regions, instead of undercutting where boundaries between classes or objects may be lost. It is much easier to merge regions together into objects or classes than to later try to split them apart. Presumably, whatever criterion would be used to split up a region could have been used originally to create two regions instead of one.

4.2 BOUNDARY POINT DETECTION

There are two traditional approaches to the segmentation problem. One can use an algorithm which finds the boundaries or points of large local change in a property (image measurement), which is called "boundary point detection," or one can try to find the areas in the image where the property (or properties) are locally homogeneous, which is called "region growing." The properties used for this segmentation could be simple grey-scale intensity, color (hue, saturation, and brightness), a measure of local texture, or some combination of these properties (and perhaps others). The type of segmentation used does not depend on the properties, although presumably the quality of the segmentation would be affected.

The main advantage of the "boundary point detection" approach is that it is a line-oriented process. The points can be processed in any order that is convenient, provided that each point in the image is processed. This means that only a small amount of memory is needed during the processing, and this method is usually very fast. The main disadvantage of this approach is that although finding the boundary points and marking them is fast and easy, combining these points into contiguous boundaries is not. In some cases, poor data quality will lead to boundaries for which some edge points are missing. This means that the boundary follower must be able to fill in the gaps without creating a lot of meaningless boundaries from noise points.

4.3 PHASE I REGION GROWER DEVELOPMENT

The main advantage of region growing is that it guarantees that the resulting regions will be made up of connected points and all boundaries between regions will, therefore, be connected. There are two main problems.

The first is that region growing is not a line-oriented process. The regions are normally grown outward in a wave-oriented fashion, which makes it difficult to access the pixels from a disk-like device. The second major problem is weak boundary points. These points are the same ones that cause unconnected boundaries in the boundary detection approach. Unfortunately, the problem is more serious in region growing. Once a region has grown through a weak boundary point, there is no turning back, so a large and perhaps important boundary can be lost because of only a few bad data points.

4.3.1 First Generation Algorithm

The region grower algorithm that was originally implemented by D. Taenzer at RADC on the PDP-11/20 has actually been developed over a number of years. The specific nature of the algorithm is perhaps best explained in terms of its chronological development. The original version of the algorithm was designed as part of a program to automatically analyze printed circuit board solder joints. The problem was to segment an image of a solder joint into areas of gloss, metal, shadow, and P.C. board. Later routines then analyzed the relative position and size of these regions in terms of the illumination used at the time the image was digitized to determine if there were any faults in the solder joint. The first segmentation scheme used was simply to threshold the image by intensity values. This was based on the assumption that there was a correspondence between the intensity at a point and what material it represented. This simple approach failed because there were large illumination gradients in these images, and in some cases a shadow area in one part of the image was at a higher intensity than a metal area in another part of the image.

4.3.2 Second and Third Generation Algorithms

The next approach was to use a simple region grower. This algorithm grew regions outwards from a specified starting point in all directions to produce a wave-like growth. The decision whether or not to include a point in the region was based on a simple intensity threshold. If a point under consideration had a neighbor (adjacent point) that was in the region being grown, and if the intensity difference between these points was less than some value, the point was added to the region. This algorithm solved the illumination gradient problem since such gradients normally have small slopes, which produce small local intensity differences; yet the algorithm still did not produce adequate results.

There were two major problems with this version of the algorithm. The first was that it had trouble labelling points with large local intensity differences. This was caused by the fact that this was a "one-pass" approach and all points had to get some sort of label. This meant that when a region grower found that a point was above the threshold in intensity difference, it simply tried another point. This boundary point, or "edge point," might eventually get into the region anyway if the intensity difference to some other point in the region was below the threshold. In other words, the one-pass approach only saw very strong edge points that were significantly different from all points in the region. This was too strong a criterion to apply to the boundary decision.

The second problem was that regions were growing through weak boundary points and were often losing large boundaries because of only one noisy point. The author refers to this as the "go problem," after the ancient Chinese game

of strategy. The goal of the game is to surround areas of a board by making boundaries on a square raster grid. One weak spot on such a boundary can lead to disaster. These problems were solved in the next version of the region grower. The new version used two passes instead of one. The first pass grew regions by accumulating points that it knew were connected and left any questionable points to the second pass, which took care of the slope and noise points. The first pass of the grower had another important feature. It was not sensitive to the local pattern of growth. In other words, before a new point was added to a region, it was determined if there was a chance that the program was growing through a hole in the strong boundary, or "bottleneck." The grower now examined a local neighborhood to see if the pixels adjacent to a point were boundary points, and refused to include a new point if there was any doubt.

At the end of this somewhat timid first pass, there were some points that were unlabelled. These were the points between regions where the local intensity differences are larger than the threshold (by definition) and some noise points in the middle of the regions. The second pass was designed to label these points in a reasonable way. It scanned through the image until it found a slope point and then examined its eight neighbors. If there were neighbors that were labelled on the first pass, the program determined if they were all in the same region. If so, it labelled the slope point as being in that region. If more than one region was represented in the neighborhood, the slope point was assigned the name of the labelled neighbor that was closest in intensity. If none of the neighbors were labelled, the point was left as a slope point. This was done to make the second pass labelling independent of

the scanning algorithm. Since there might be some points left unlabelled, this process was repeated for the whole image until all edge points were labelled.

This two-pass algorithm worked reasonably well on images that contained large homogeneous areas, but had great difficulty with highly textured images. A highly textured area was split up into many small meaningless regions that were difficult to merge together. Also, since each region had to have a unique name, a large number of names were needed for very small images (64 by 64 pixels). This meant that either a large amount of memory had to be devoted to the pixel names, or the region grower would run out of names to use on some images.

4.3.3 Current Generation Algorithm

The first attempt at solving the problem of small regions was to label them as "edge" points during the first pass of the region grower. Then these points would be automatically labelled in the second pass according to the intensity of the pixels. This approach had the effect of removing the naming problem, but led to another difficulty, which has been called the "two-lake problem" after an image on which it was first demonstrated. In an image containing two strong homogeneous regions separated by a highly textured area, such as two lakes with visible trees between them, the points in the highly textured area were merged into one of the two lakes. The net effect was to create one boundary between the two lakes that did not correspond to the shore of either one, but was instead simply a line between the two closest shores. The next approach was to label all the "edge" pixels in all of the small regions with the same "reject" name. This had the advantage that the points

from the small region were not relabelled during pass two, but still left these small regions after the processing.

The texture and naming problem were solved by putting in three more passes into the region grower, and adding an area threshold to define the size of the smallest meaningful region. After a region is grown in pass one, the new algorithm checks to see if the region is larger than the size threshold. If not, it relabels the points in the new region to a special "reject" name. All such small regions are thus labelled as "reject" regions after the first past. The second pass is unchanged, except now some of the edge points may be labelled as reject points.

There is now a third pass in the algorithm which grows regions out of reject points in a similar manner to the way regions are grown in pass one. It scans for a reject point and examines its neighbors to see if they are reject points. Any reject neighbors are included in the new region and given a new unique region name. The net result of this growth is that adjacent regions will now be merged together into larger regions which by design are regions of some degree of texture. The merge criteria is now the size of the original region, which makes this simpler than most region-merging algorithms (such as the one used in pass five).

The main difference between pass one and pass three is that pass one uses intensity information to determine if two pixels belong in the same region while pass three uses the region name information generated by the first two passes. While pass three is growing the reject regions, it keeps a list of the name of the reject region and all other regions that are adjacent to it. These adjacent regions are large regions grown during pass one.

The fourth pass simply goes through the image calculating the area and average intensity for all the regions. This information is used in pass five. The last pass goes through the list of reject regions and neighbors created during pass three. For each reject region grown in pass three, pass five determines from the statistics table if the area of the region is now larger than the size threshold used during pass one. If it is, the reject region is left unchanged. If it is still too small, then pass five determines which of its neighbors is nearest in average intensity to the small reject region and merges the reject region into that neighboring region.

At the end of this processing, each pixel in the image has a region name associated with it. These regions are guaranteed to be made up of connected pixels, and each region is larger than the specified size threshold. This means that each region is now large enough to be meaningful in terms of the context of the image. This size threshold method for dealing with fine texture has the advantage that the algorithm does not have to know the exact size of the texture elements; only that they are smaller than some specific area threshold. This area can be easily related to a specific size in the real world by using information about the scale or magnification used during image digitization. The net result of all of this is that the region grower is no longer particularly sensitive to either the intensity or area thresholds used during the processing.

4.3.4 Commentary on Phase I Region Grower

The motivation for presenting this chronological development has been to show that the apparent complexity of the present algorithm has really developed from attempts to solve problems inherent in the earlier versions of

the algorithm. The local pattern of growth analysis was added to solve the "go problem." The addition of the concept of reject region, and the growth of such regions based on region name information, was an attempt to deal with the problem of texture. This approach to texture is perhaps one of avoiding the issue in some sense, since the information about individual texture elements is lost during the processing. It does have the feature that it finds the more global features present above the level of the texture, however, and is using a threshold based on the minimum area of meaningful features to separate out the texture elements.

The present version of the region grower algorithm differs from the "classical" pattern analysis approach in several important ways. The first is that statistical pattern recognition is an inherently local process involving each individual pixel and perhaps some pixels in a neighborhood immediately around it. The results of such analysis often produce regions of the different categories that are not made up from spatially connected points. This leads to a boundary-finding problem similar to that encountered in the boundary point detection algorithms. This is an unavoidable result of local processing which has no way of using more global information about the topology of the image.

Another problem with certain statistical pattern recognition approaches to this problem is how they deal with highly textured images. One way of dealing with such images is to reduce the resolution of the data to the point where the texture elements are no longer uniquely recognizable. This approach is reasonable for many types of images, but suffers from the problem that the resolution must be adjusted to one particular texture element size. This

information may not be known before processing the image and may, in fact, vary over different parts of a single image.

Region growing and statistical pattern recognition do not need to be considered as alternatives to each other in image processing. It may be possible to use the information generated during region growing to form feature vectors based on regions instead of pixels. This would lead to two interesting improvements. First of all, the classification boundaries would already be defined and guaranteed to be connected and contiguous. Secondly, the generation of a feature vector from a combination of many pixels should lead to a more accurate classification of the image. This might also allow more accurate classification of highly textured areas in some images.

4.4 PHASE II REGION GROWER DEVELOPMENT

The first version of the region grower was written in DEC assembler and run on the DEC 11/20. With the installation of an 11/70 using the UNIX operating system, it became necessary to rewrite the region grower in such a way as to interface properly with the UNIX operating system. During this phase of development it was decided to investigate the possibilities of extracting regions from images regardless of the size of the image; and the possibilities of using features other than intensity when extracting regions. Of the many features which have been used in the past in other research for region extraction, texture measurements were of particular interest and, therefore, investigations were initiated to probe the feasibility of using texture to discriminate regions in an image.

4.4.1 Implementation under UNIX

Implementing the region grower under UNIX involved two efforts. First, all the programs were rewritten using the C language. The use of a higher level language such as C has many advantages including ease of programming, ease of documentation, and enhanced readability of the resulting code. The original version of the region grower was quickly and easily rewritten in C, however, it only concerned itself with small images. Fortunately, the use of C and UNIX allow easy and efficient modifications of code and the programs could be quickly rewritten to use full-sized images.

The second effort involved insuring that the region grower was compatible with the AFES effort. Basically this meant that the region would output a mask of the region grown similar to the mask produced by the get_mask program. This compatibility allows the region grower to be used with trainers and classifiers built under AFES.

4.4.2 Advantages of Using Random Access Display Refresh Memory

Originally, the region grower program was designed to use 63 x 23 character buffers internally in the 11/70 to hold the input (image), the output (names), and an intermediate data buffer (fill) which was used to hold results while edge and noise points were being filtered out of the names buffer. However, in designing the program to grow one or more regions in a full size image it was quickly realized that neither the 11/34 nor 11/70 could hold all the data buffers in core simultaneously. Since region growing is a random process it is necessary to keep all the data buffers available, which implies keeping two full-sized images in core. Therefore, it was decided that

the memory associated with the displays themselves should be utilized. That utilization was carried out as follows on the black and white displays.

First of all, the image plane itself could be used as the input buffer (image), and all other buffers were defined in the overlay planes. The output buffer (names) is loaded into overlay planes 0 and 1 which generate 4 possible values (0,1,2,&3) in the buffer; where 0 implies an unknown point, 1 is an edge pixel, 2 is defined as a region point, and 3 is undefined presently. The intermediate buffer (fill) is loaded into planes 2 and 3; and is utilized similar to the names buffer in that it also generates four values. Value 0 is defined as edge, 1 is called unknown, 2 is region, and as before, 3 is undefined.

A series of macros and an initialization routine designed by J. Chmill are used to randomly access the buffers. The DISP macro allows the program to access any specified pixel and init_disp initializes the DeAnza display control registers so that the DISP macro may be used.

Building the region grower on the color display was achieved simply by using the red image plane as the image array, the green image plane as the names array, and the blue image plane as the fill buffer. The random access macros did not require modification.

4.4.3 <u>Thoughts on Exploiting Texture for Region Growing</u>

It has been mentioned earlier that more than one kind of feature data may be used to extract region information from an image. One of the most widely recognized and well-investigated is the use of texture, therefore, it was

decided to investigate the possible use of texture in region growing. Also, the possible use of more than one feature in region growing was researched.

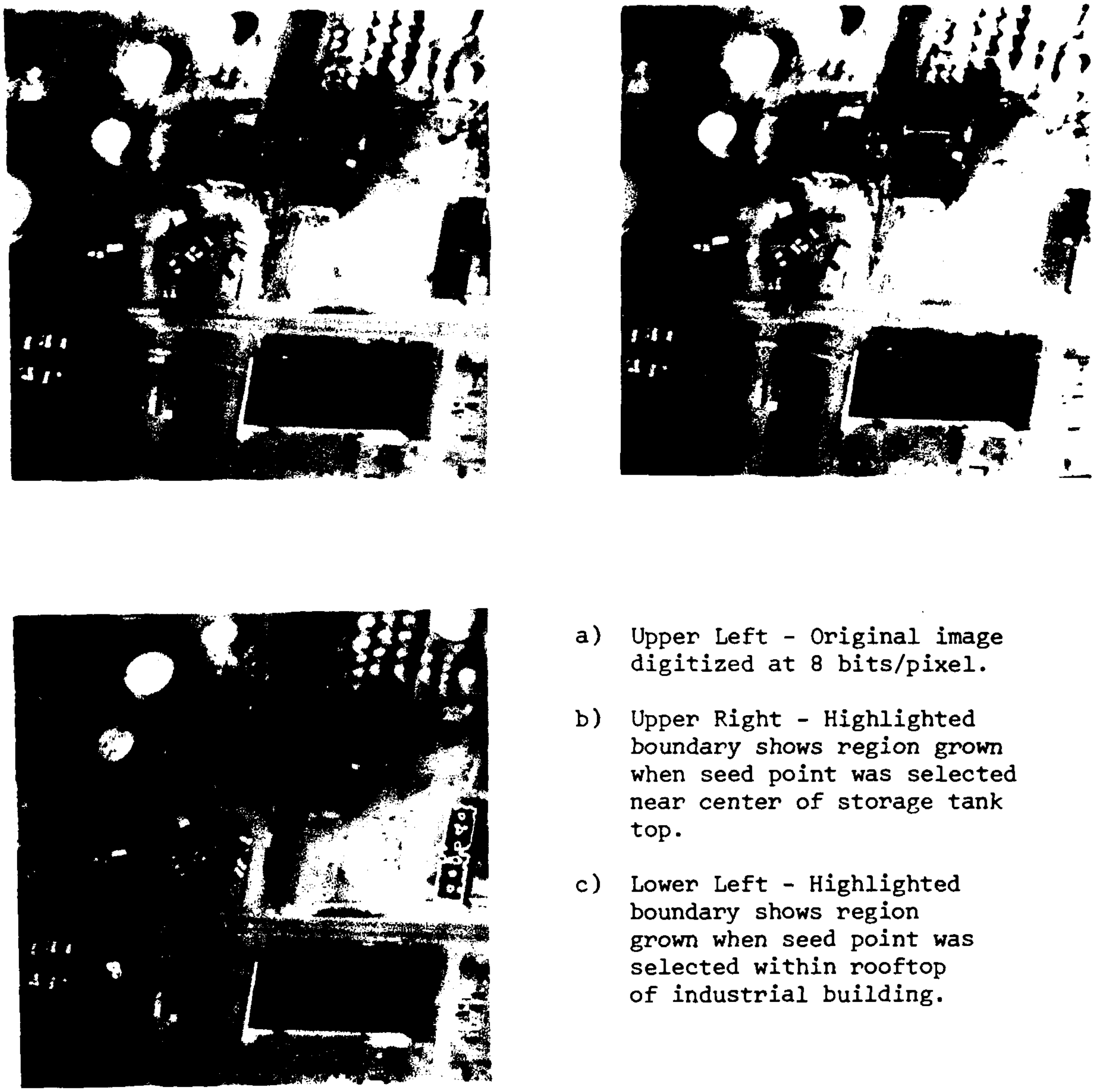
One of the first questions that arose was what kind of texture measurements that would be applicable to the system in use. Two efforts were initiated; one involved the implementation of Haralick's texture measures, and the other was concerned with the use of Hsu's measures.

Since the region grower uses the display memory for data buffers it was necessary to write the texture programs in such a way that input and output were both images. This was accomplished by simply determining the largest and smallest texture results and scaling the data into values between 0 and 255. The resultant image was simply loaded into the appropriate image buffer and used by the region grower.

Finally, if growing on more than one feature is desired, it is possible to merge several images into one before using the region grower. There are at present two methods available for merging images. First, there is a simple vector merge called merge_if, and there is a principle parts analysis routine which uses multivariable statistical analysis to perform the merge.

4.5 EXAMPLES OF REGION GROWING

In this section several pictorial examples of regions extraction by region growing are presented. These are illustrated in Figures 4-1, 4-2, and 4-3 on the following pages.

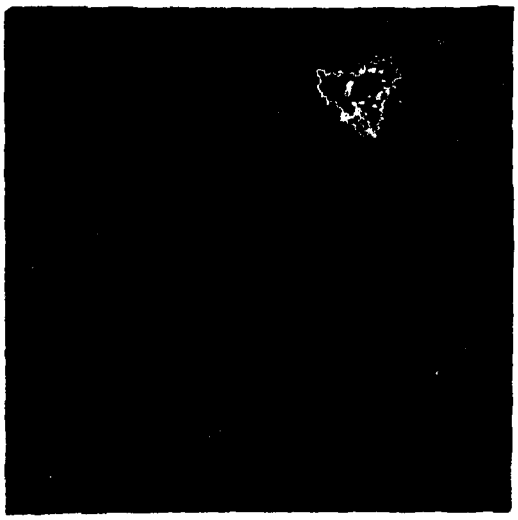


a) Upper Left - Original image digitized at 8 bits/pixel.

b) Upper Right - Highlighted boundary shows region grown when seed point was selected near center of storage tank top.

c) Lower Left - Highlighted boundary shows region grown when seed point was selected within rooftop of industrial building.

Figure 4-1 Example of Region Growing on High-Resolution Monochrome Aerial Photography. Images were photographed from DeAnza (512 x 512) Color Display.



a) Upper Left - Original image digitized at 8 bits/pixel.

b) Upper Right - Highlighted boundary shows region grown when seed point was selected within cultivated field.

c) Lower Left - Highlighted boundary shows region grown when seed point was selected within area of vegetation.

Figure 4-2 Example of Region Growing on Moderate-Resolution Monochrome Aerial Photography Images were photographed from DeAnza (512 x 512) Color Display.

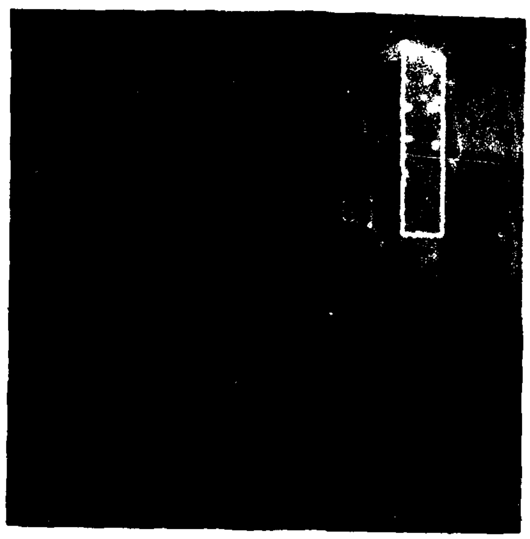


Figure 4-3 Example of Region Growing on Near-Infrared Photography. Images were photographed from DeAnza (512 x 512) Color Display. Top photograph shows the original image digitized at 8 bits/pixel. Bottom shows highlighted boundary representing area grown when seed point was selected within building rooftop.

SECTION 5

CONTRAST RESTORATION OF AERIAL IMAGERY

5.1 ACKNOWLEDGEMENT

The information presented in this section describes an approach for removing atmospheric haze effects for contrast restoration of aerial imagery. Software algorithms to implement these procedures were developed and are described herein. The presentation given here, as well as in Appendices A, B, and C, was written by:

L. W. Crouch

J. D. Leonard, Jr.

Dynamics & Environmental Branch

Reconnaissance & Weapon Delivery Division

Air Force Avionics Laboratory

Wright-Patterson AFB, Ohio

5.2 APPROACH TO THE PROBLEM

5.2.1 The Atmospheric Problem

The natural process by which the earth receives energy from the sun and reflects or reradiates it back into space must be considered in the analysis of imagery collected by remote sensing techniques. As solar energy (light) enters the upper atmosphere, the molecules of air and particles of dust, etc., cause a portion of the energy to be scattered in all directions as non-image forming light (haze), a second portion of the energy is absorbed in the atmosphere, and the remainder of the light reaches the ground to illuminate the scene. The ground scene absorbs some of the energy and

reflects the remainder in accordance with the reflectance characteristics of the specific materials of the scene. The reflected light is transmitted through the atmosphere (again undergoing scatter) to the sensor as image forming light. The haze and image light ultimately arriving at the sensor have been attenuated to a value less than 100 percent of the available solar illumination. Most of the energy loss is accounted for by absorption along the entire transmission path.

The presence of the haze within the field of view of the airborne sensor reduces the aerial contrast ratio of the terrain image in a mathematically predictable manner. However, equations describing these relationships generally are not practical to use since they require the definition of all the specifics of the entire process at that instant of time (i.e. haze, moisture, aerosols, to name but a few). For additional details of the atmospheric considerations, see Appendix A. The effects of atmospheric haze on aerial contrast, although systematic, vary widely as a function of the mission and illumination geometry. This means that each image will be consistent mathematically but will appear differently as illumination and sensor angles are changed. The analytical power of the techniques used in the haze model, described later, do permit the reconstruction of the atmospheric factors and therefore, the removal of the effects of haze.

5.2.2 Use of the Haze Model Equations

The practical use of target reflectance and contrast data, a longstanding problem, now has a workable solution using the various atmospheric haze models. One such model computes the haze (NA), inherent image forming light (N_{100}), path transmittance (τ), and the ratio ($Na/N_{100}\tau$) x 100 which is

identified as the "C" factor. These terms are used to compute the aerial contrast ratio by:

$$C_r = \frac{R_b + \dfrac{N_a}{N_{100}\,\tau} \times 100}{R_t + \dfrac{N_a}{N_{100}\,\tau} \times 100} \qquad (1)$$

and

$$C_r = \frac{R_b + C}{R_t + C} \qquad (2)$$

and

$$C = \frac{N_a}{N_{100}\,\tau} \times 100 \qquad (3)$$

where C_r = Aerial contrast ratio

R_t = Reflectance of the object in percent

R_b = Reflectance of the background in percent

C = "C" Factor in percent

N_{100} = Inherent Radiance from a perfect diffuse 100% reflectance target ($W/Sr/cm^2/cm$)

τ = Optical path transmittance (decimal)

N_a = Optical path radiance ($W/Sr/cm^2/cm$)

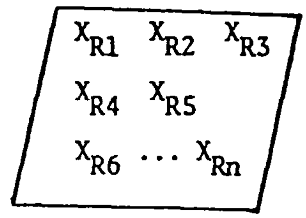
The key point is that use of the haze model techniques and equations 1, 2, and 3 permits any contrast analysis to include all the illumination, geometric, and haze parameters embodied in the model. This capability provides values of energy versus wavelength for additional analysis, if desired.

The problem addressed by this work effort used the above equations to define the atmospheric effects upon the photographic image and, in turn, to compute the adjustment in the image contrast to remove the atmospheric effects. Briefly, two high altitude photographs of the same area were selected for demonstration of the technique. The methodology outlined for the image processing procedure starts with digital image data of the photograph, the D Log E curve and a set of equations to convert the image density through the D Log E curve into exposure energy. From this point, the aerial contrast can be calculated and used in conjunction with known target and background reflectance values to compute the value of "C." Within a given scene, near vertical and exclusive of clouds, the atmospheric effect is considered to be a constant. This condition would then permit computation of the reflectance for every pixel in the scene based upon the other three terms of the equation being known or measured. The original planned approach was to use equation (2) and the above technique to compile the contrast adjustment data. However, during the course of the program a simplified approach was found that was mathematically rigorous.

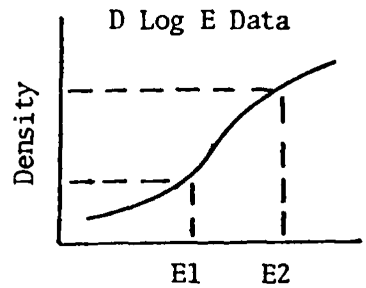
The technique is illustrated in Figure 5-1. Briefly, five operations are used to convert the degraded photograph to one without haze. Given the reflectance values of known ground objects such as fields, large roof tops, runways, etc., these can be used in a linear regression analysis along with their exposure values (AE) to develop the coefficients (m) and (b) of:

Figure 5-1 Contrast Adjustment Technique

Step 1. Select objects of known reflectance - R, R_2, ..., R_n and measure their densities - D_1, D_2, ..., D_n

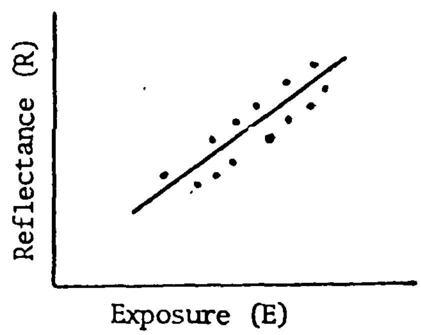


Step 2. Convert the measured density values of known reflectance to exposure values (E_1, E_2, ..., E_n)



Step 3. Perform a regression analysis of known reflectance (R_n) and exposure values (E_n) to determine the coefficients m and b for

$$R = m\,E + b$$



Step 4. For each density value of the D Log E, calculate the E value as in Step 2 and calculate the reflectance using the equation of Step 3. This provides a look-up table of density to reflectance values.

Density	E	Reflectance

Step 5. For each pixel in the photograph, use the density value to look up the reflectance value (0 to 1.0) to build a new picture that is free of atmospheric effects.

Reflectance	Display Conversion Factor	Display Image
0	0	0
1.0	256	Max.

$$R = mAE + b \qquad (4)$$

where R = Reflectance of object

AE = Exposure value in energy units

This provides the basic equation that can be used to convert the remainder of the unknown image data to reflectance values through use of a look up table which becomes a dramatic time saver in the program. The reflectance value provides the numerical description of the terrain or object surface that is free of effects of the atmosphere and is, therefore, the numerical value used in image reconstruction. The reflectance values ranging from 0 to 1.0 are used to define the image density or display brightness as the final output to the contrast adjustment subroutine.

5.3 DETAILED PROCEDURE

5.3.1 Development of Equation

The light which exposes aerial imagery is from two sources: that reflected from the target and that reflected from the haze in the atmosphere. A term relating these two values, called the "C" factor (See Equation 3), is important because under the proper conditions it can be assumed constant over the entire image. These conditions are: that the haze is even over the scene, and that the scene is evenly illuminated as viewed by the imaging sensor. Normally this would require the scene to be viewed at a near vertical angle, however, oblique scenes can be used where they meet the above requirement.

When these conditions are met, the contrast ratio

$$Cr = \frac{A_{E1}}{A_{E2}} \qquad (5)$$

(where Cr is the contrast ratio and A_{E1} and A_{E2} are the absolute aerial exposure values of two objects in the scene) can be expanded to

$$Cr = \frac{N_{100} R_1 \tau + Na}{N_{100} R_2 \tau + Na} \qquad (6)$$

and using equation (3) can be reduced to

$$Cr = \frac{R_1 + C}{R_2 + C} \qquad (7)$$

where R is the reflectance value of an object (this is equivalent to equation 2). Solving for C yields

$$C = \frac{R_1 - C_r R_2}{C_r - 1} \qquad (8)$$

which combined with equation (5) yields

$$C = \frac{R_1 A_{E2} - R_2 A_{E1}}{A_{E1} - A_{E2}} \qquad (9)$$

With equation (9) and any two objects of known reflectance and measured aerial exposure, the "C" factor for the scene can be calculated.

Since the "C" factor is assumed constant over the entire scene, the reflectance of any other object in the image can be calculated using equation (9) in the form

$$R_D = \frac{C(A_{ED} - A_{EK}) + R_K A_{ED}}{A_{EK}} \qquad (10)$$

where R_D = the reflectance of the third object

A_{ED} = the measured absolute aerial exposure of the third object

and K is either 1 or 2, meaning the use of objects 1 or 2.

Equation (10) can be rewritten as

$$R_D = \frac{(C + R_K)}{A_{EK}} A_{ED} - C \tag{11}$$

which is in the form of a straight line

$$R_D = mA_{ED} + b \tag{12}$$

where m = the slope

b = the R_D intercept

Since equation (10) can be written as a straight line, linear regression can be used to calculate the constants "m" and "b" in equation (12). With linear regression, a large number of calibration points can be used to calculate an average relationship between exposure and reflectance. This averaging capability is important since the reflectance of most objects varies depending on such factors as spectral characteristics surface texture, sun angle, moisture content, and the age of the object. Therefore, the exact ground reflectance of a class of objects cannot be tabulated, and if only two objects are used for calculating a "C" factor, the error almost certainly would be noticeable. However, using several objects, including objects of low, medium, and high reflectance, an accurate and consistent calibration can be obtained as demonstrated by the test described in the following section.

5.3.2 Aerial Contrast Restoration Procedure

The process of contrast restoration begins with a digitally sampled image. The specific area selected for restoration should conform to the haze and illumination constraints discussed previously. Where deviations from

these constraints are present, such as clouds and non-uniform haze, the restored image will contain the residual excess of the clouds and haze.

The first step of the procedure (refer to Figure 5-1) is the definition of a number of object areas of known reflectance for use as calibration points. These areas must be void of specular reflection, must be of uniform density, and must be relatively large with respect to the digital sampling and system resolution to insure that their film density level was altered only the the atmosphere. Typical areas that normally would be suitable for calibration points include runways, fields, large roads and warehouse roofs. A list of some suggested calibration objects and their respective reflectance values is provided in Appendix C. By looking at a print or display of the scene, the approximate boundaries of each calibration area are first defined. Then each of these areas is enlarged to the point that the individual pixels can be identified so that the exact cartesian boundaries of each calibration area can be determined. The average density within the boundaries of each calibration area is then computed using a special averaging program listed in Appendix B.

In the second step of the procedure, the relative exposure for each calibration area is computed. This is accomplished by first entering the D Log E curve data and the average density values of the calibration areas into the program. The program converts each density value into log exposure using the D Log E curve and then takes the antilog of the log exposure to convert it to relative exposure using

$$E = 10^{\mathrm{Log}(E)} \tag{13}$$

where E = relative exposure of the film.

Since relative exposure (E) is directly proportional to absolute aerial exposure (A_E), the linear relationship of equations (10), (11), and (12) can be extended to include a linear relationship between relative exposure and ground reflectance. Now, since this same linear relationship exists for both the calibration points and the total scene image, a significant computational reduction is realized by using relative exposure (E) instead of absolute aerial exposure (A_E).

In the third step of the procedure the linear regression coefficients relating the average aerial exposure of each calibration area to its expected reflectance are calculated. At this point a relationship exists between relative aerial exposure and ground reflectance and also between measured film density and relative exposure using the D Log E curve. Since there are only a limited number of possible quantization levels of density, a table relating these density levels to ground reflectance is calculated as shown in Step 4. Density has a range from the toe to the shoulder of the D Log E curve which is divided into increments of 0.01 units of density. For each of these density values the corresponding relative exposure is calculated, and is then used to calculate the corresponding ground reflectance. This is stored in a table of reflectances using the index of 100 times the density value. All table values which correspond to density values less than the D Log E curve toe are given a value of reflectance corresponding to the toe of the curve. Similarly all table entries above shoulder of the D Log E curve are given the reflectance corresponding to the shoulder value. Therefore, for any measured density value in the range 0.00 to 3.00 there exists a corresponding calibrated reflectance value in the table.

The final step (5) of restoration is the table look up conversion of each density pixel of the scene to a reflectance pixel. The software does this using a tape-to-tape conversion. The resulting tape of reflectance pixels is then used with a suitable image reconstruction device to generate the contrast restored picture. The actual image can be viewed on a display device or as hard copy on film.

5.4 THE TEST OF CONTRAST RESTORATION BY SCENE REFLECTANCE CALIBRATION

5.4.1 Introduction

To demonstrate the capability of contrast restoration based upon scene reflectance calibration techniques, two aerial photographs of the same scene were used. The first (Figure 5-2) was taken on a clear morning representative of light haze and the second (Figure 5-3) on a morning with a layer of haze representative of the effects of heavy haze (These photographs will be referred to as low haze and high haze throughout this report). Each photograph includes the Hawthorne Airport near Los Angeles, CA, and each was exposed at nearly equal aspect angles. The objective of this test was to process each scene independently and verify that the resulting images are nearly equal.

5.4.2 Microdensitometer Scanning

Each scene, as measured on the negative, was 9.5mm by 3.3mm. In order to eliminate the speckled effect caused by under sampling an image, the scanning spot was made about two times the distance between samples. This improves the tonal quality of the image, but softens the edges. The sampling was done through a 73 micron circular spot with an intersample distance of 30 microns.

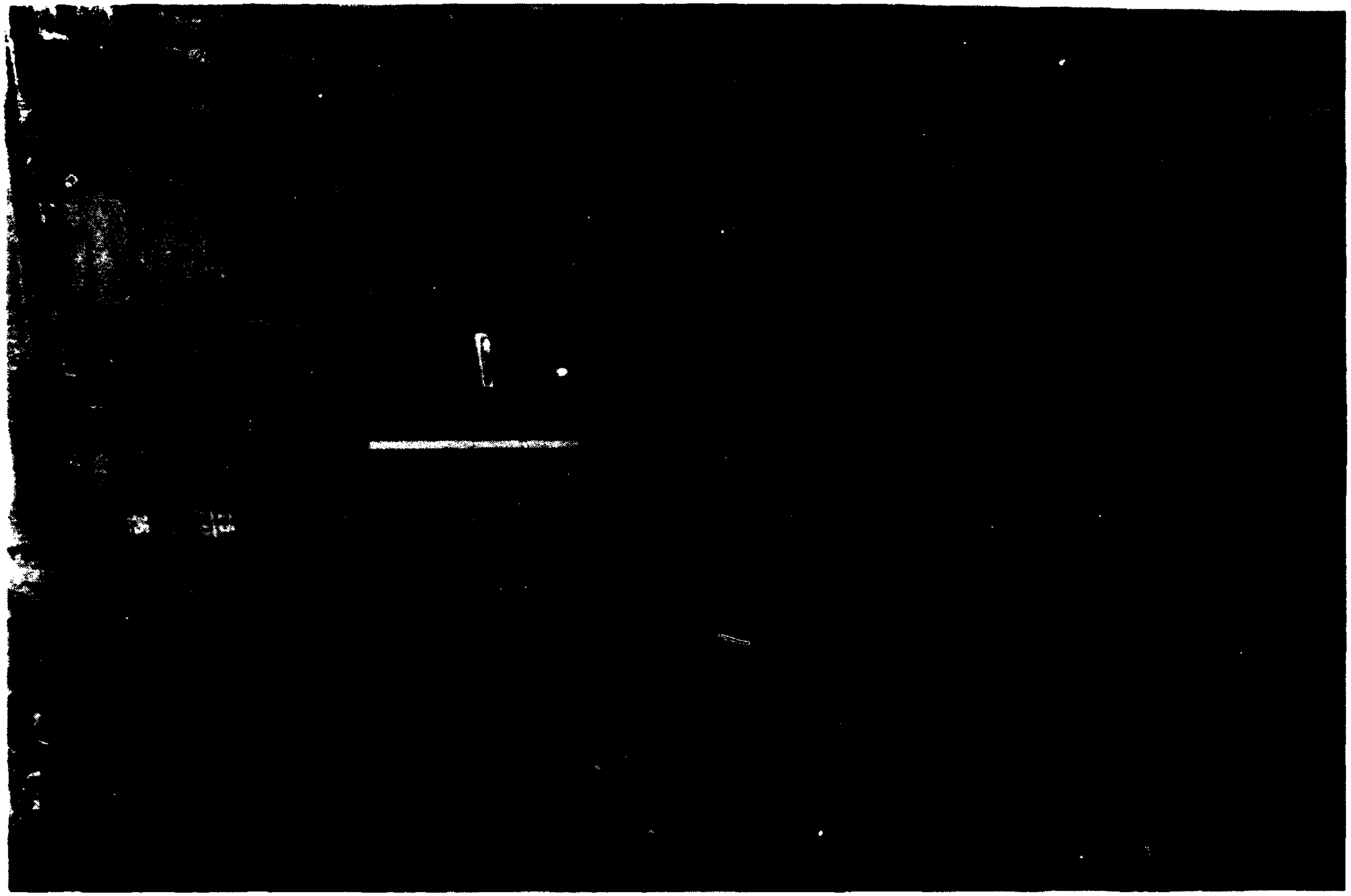


Figure 5-2 Low Haze Photo of Hawthorne Airport

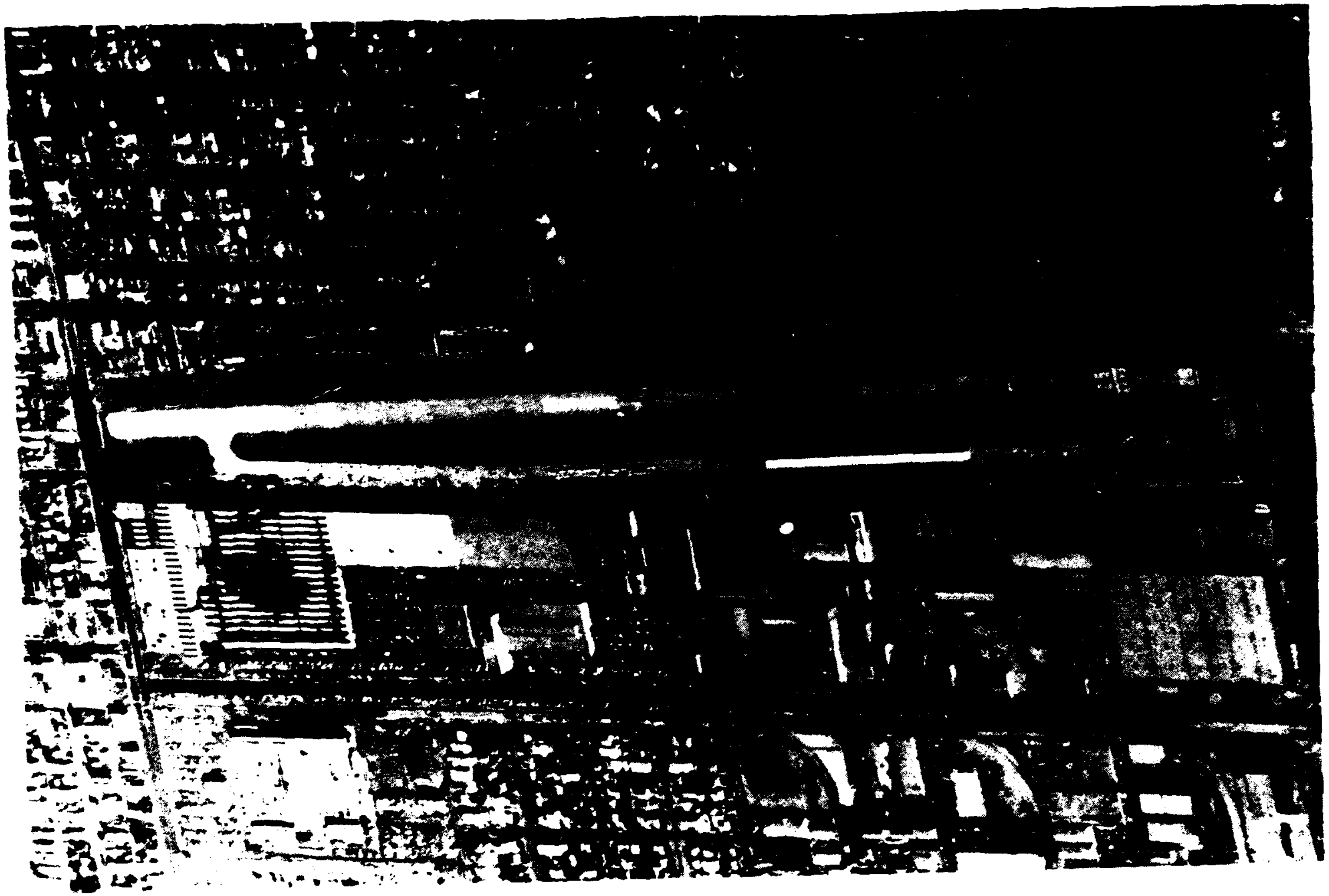


Figure 5-3 High Haze Photo of Hawthorne Airport

Thus the image was divided into 1100 pixels per scan and 316 scans. Each scan was further divided into 11 blocks of data with 100 points per block because of the microdensitometer's block size limitations. The scanning rate of the microdensitometer was set at its maximum value of 25 millimeters per minute for a total scan time of about 7 hours per scene. (Note: This scan speed is an inherent characteristic of this microdensitometer and not necessarily representative of other densitometers). In addition, for each image the visible steps of a 21-step exposure wedge were scanned using the same microdensitometer, to eliminate instrument calibration errors, and then averaged for each step. These density values were then plotted (Figures 5-4 and 5-5) versus their corresponding relative exposure values, yielding each scene's D Log E curve.

In the first test of the contrast restoration of each scene, twelve known objects were designated as reflectance calibration areas (Step 1). These included five types of objects: grass, dirt, blacktop, runway, and white paint. They were each scanned and averaged for use as calibration areas in Step 3 of the test.

It should be remembered when studying the resultant computer processed images that the objective of this test was not to improve or maintain resolution or to improve the appearance of the objects in the scene, but to restore the <u>proper</u> contrast to the scene. However, restoration of contrast does improve the appearance of low contrast images. In order to maintain the spatial resolution of the objects of the scene, both the intersample distance and spot size would need to be finer than was used. This would require

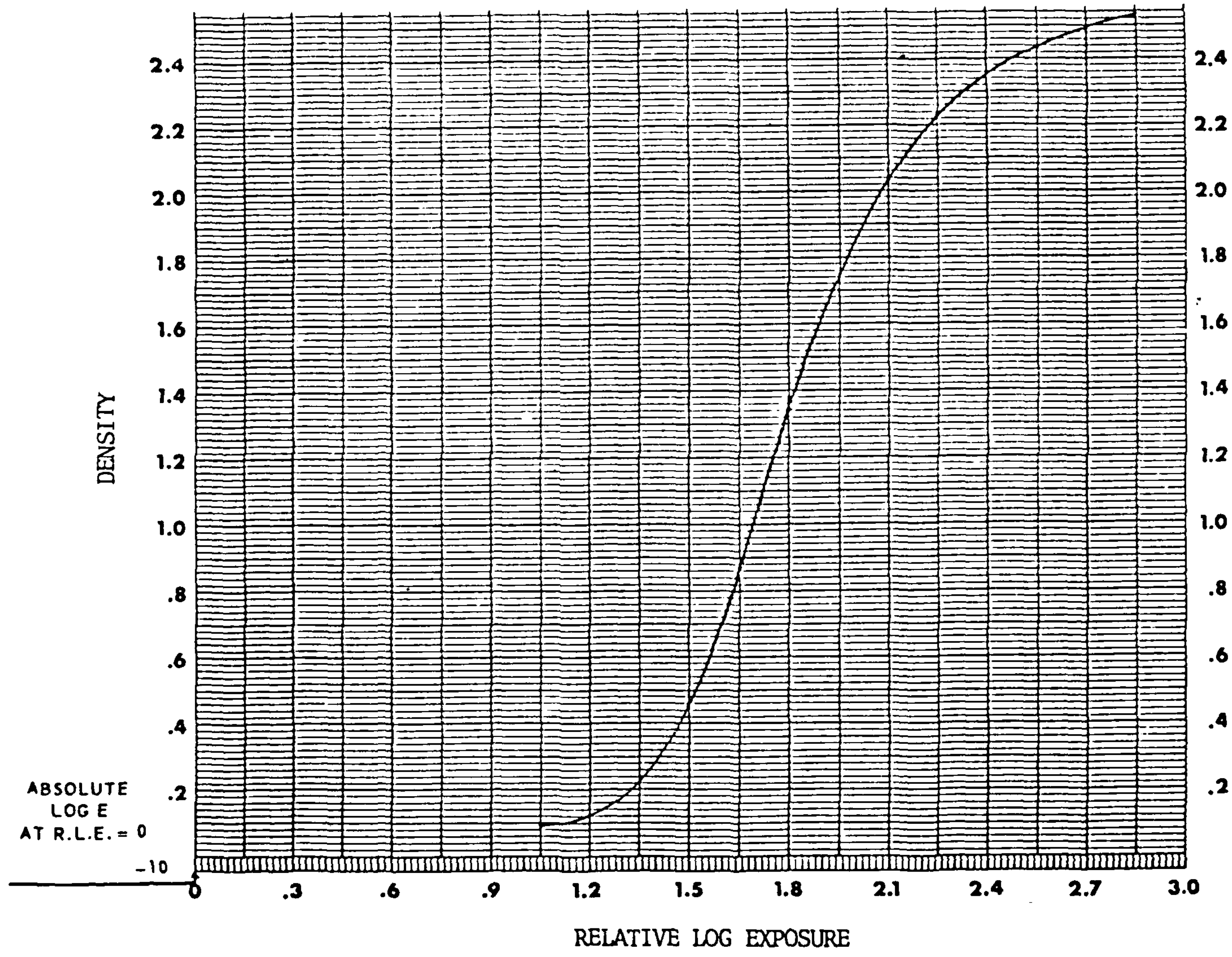


Figure 5-4 D LOG E CURVE OF HIGH HAZE FILM

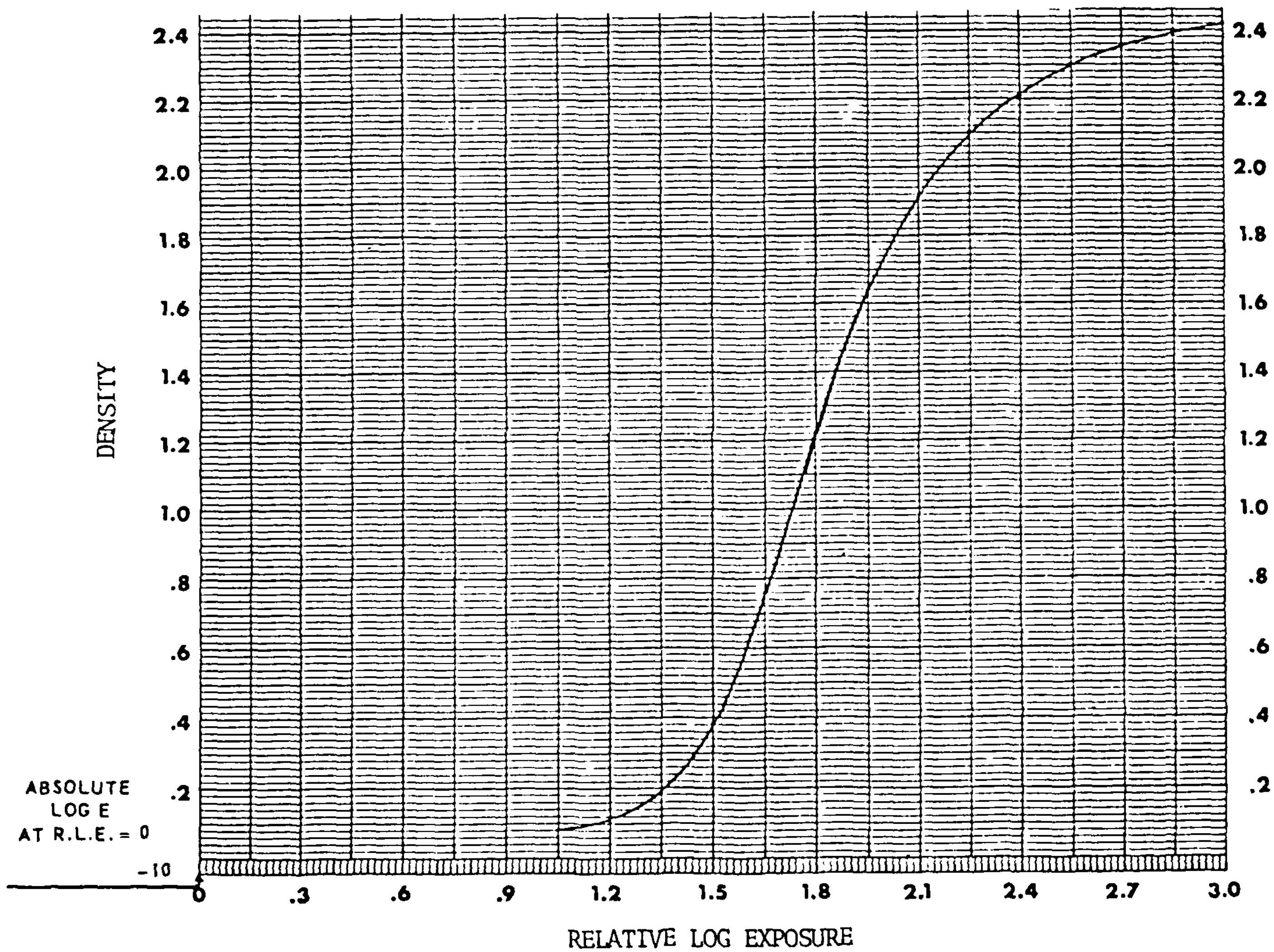


Figure 5-5 D LOG E CURVE OF LOW HAZE FILM

excessive microdensitometer scanning time using this particular microdensitometer. It should be noted that both contrast enhancement and resolution improvement could be made to the image.

After the images were scanned, the unprocessed digital density data were printed as a photograph (Figure 5-6). These show the extreme difference in contrast of the two pictures. The distributions (Figures 5-7 and 5-8) of these, two images very effectively demonstrate the effects of haze. The haze reduces the contrast of the scene as shown by the narrow distribution of Figure 5-7 as compared to Figure 5-8. It also increases the brightness as shown by the shift of the mean of the data in Figure 5-7 to a higher density level as compared to Figure 5-8. This is caused by the increase in non-imaging light reflected from the haze into the image. With the proper use of the contrast restoration technique, the resulting reflectance distributions should be similar.

5.4.3 Processing of the Test Images

Each image was processed using two sets of calibration data. To compile the first set, each of the 12 calibration objects was scanned individually by the microdensitometer in addition to being scanned during the raster scans of the total image. The second set was compiled directly from the raster scans of the image. As can be seen in Figure 5-9, the results of the first method are obviously not favorable since the apparent reflectances of the two scenes (after processing) are not equal. This is further demonstrated by the differences in their density distributions (Figures 5-10 and 5-11). The mean reflectance of the whole scene (after processing) of the clear picture is 0.27 and of the high haze picture is 0.47. The standard deviation of reflectance

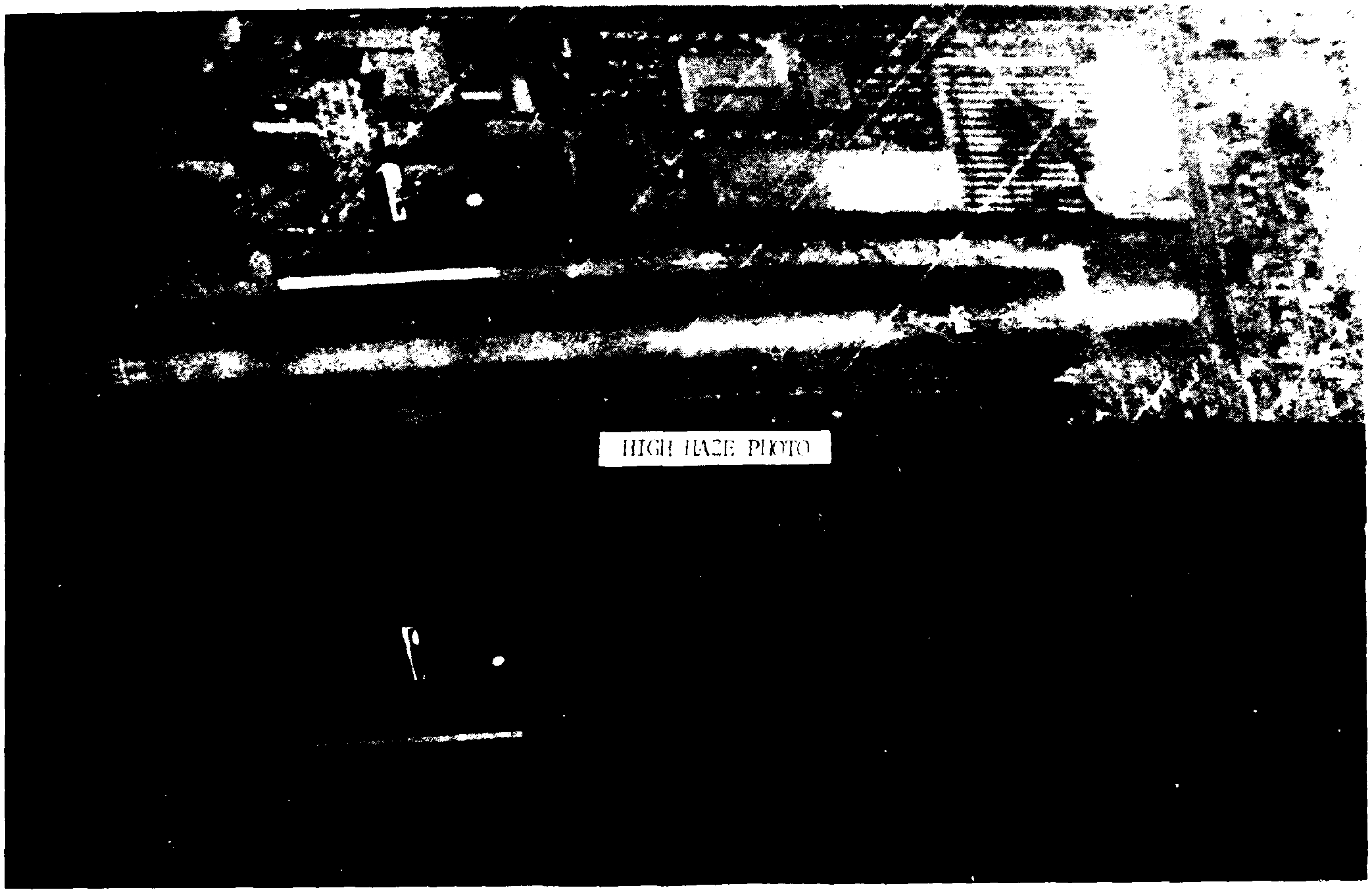


Figure 5-6 Unprocessed Density Computer Photographs (High Haze and Low Haze)

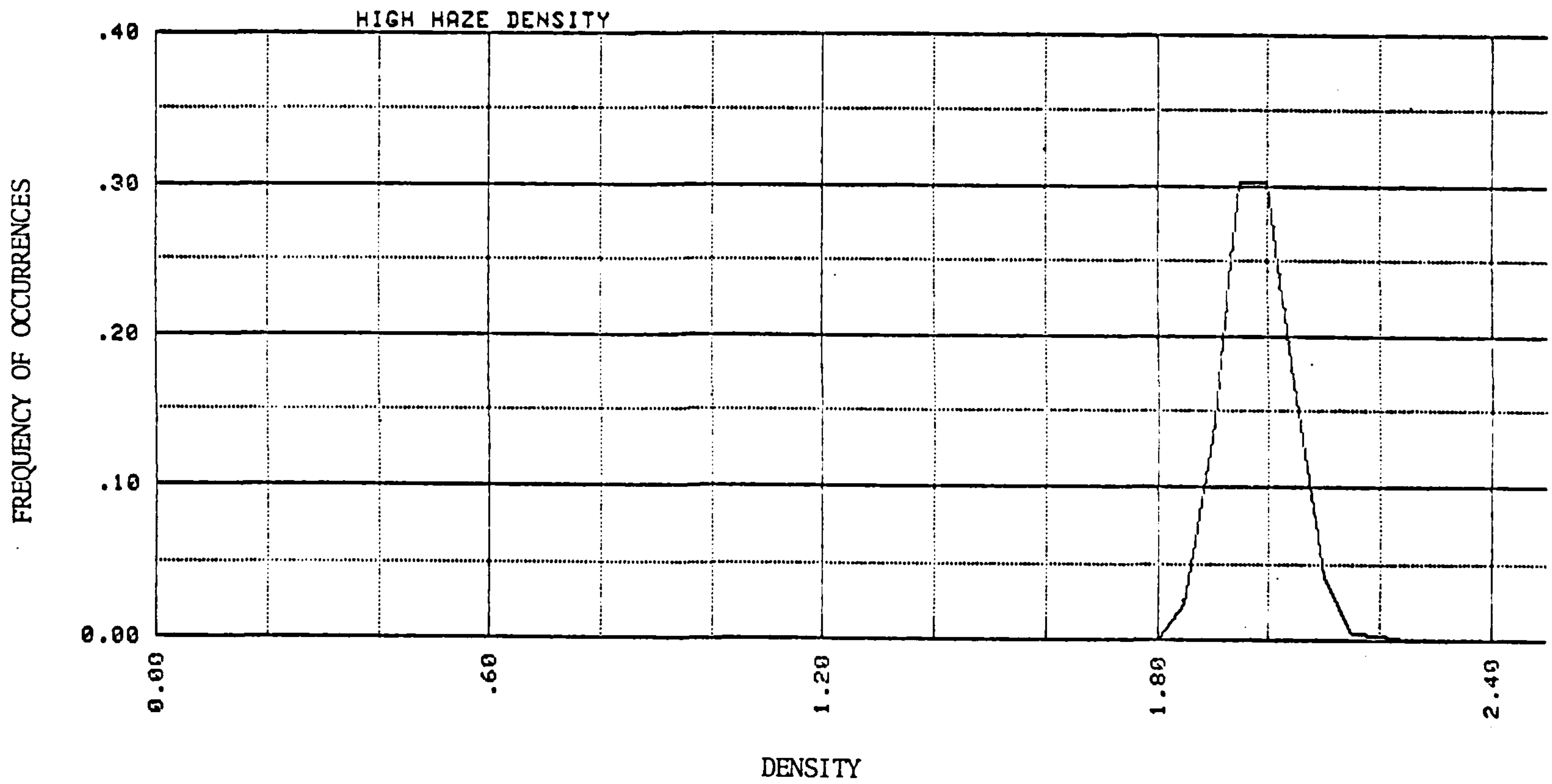


Figure 5-7 HIGH HAZE PHOTOGRAPHIC DENSITY DISTRIBUTION

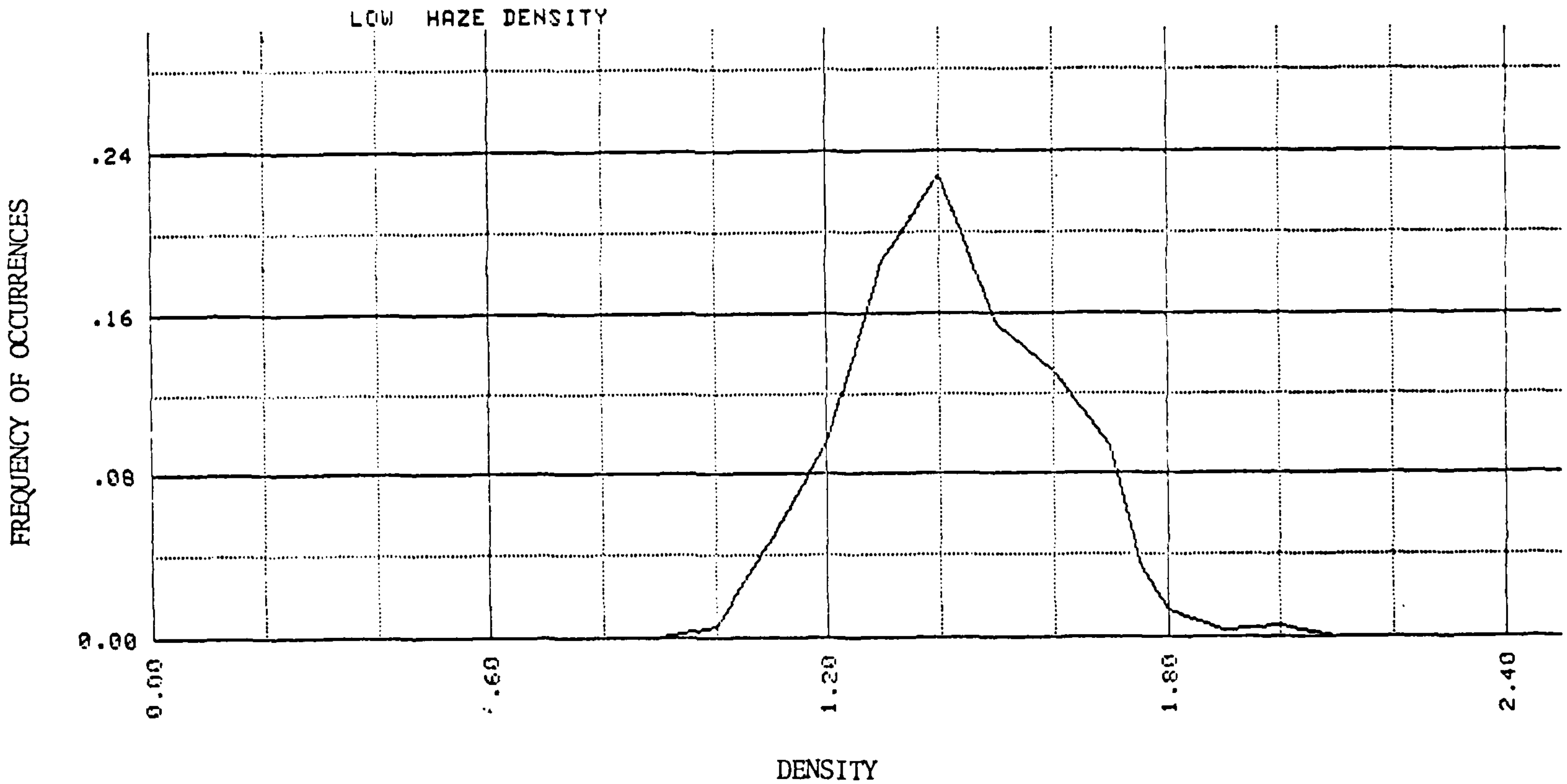


Figure 5-8 LOW HAZE PHOTOGRAPHIC DENSITY DISTRIBUTION

Figure 5-9 First Unsuccessful Attempt at Contrast Restoration (High Haze and Low Haze)

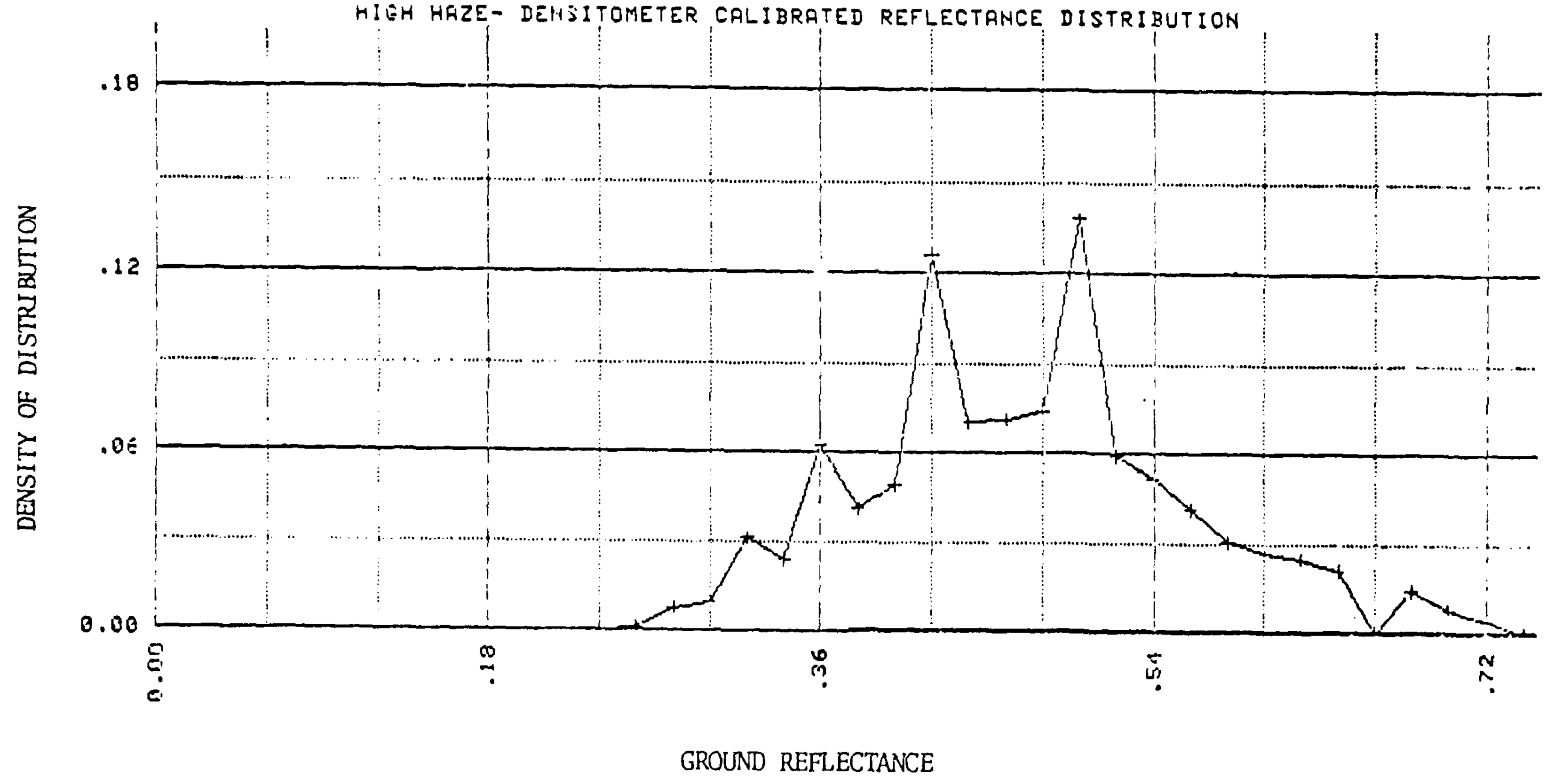


Figure 5-10 GROUND REFLECTANCE DISTRIBUTION OF HIGH HAZE IMAGE

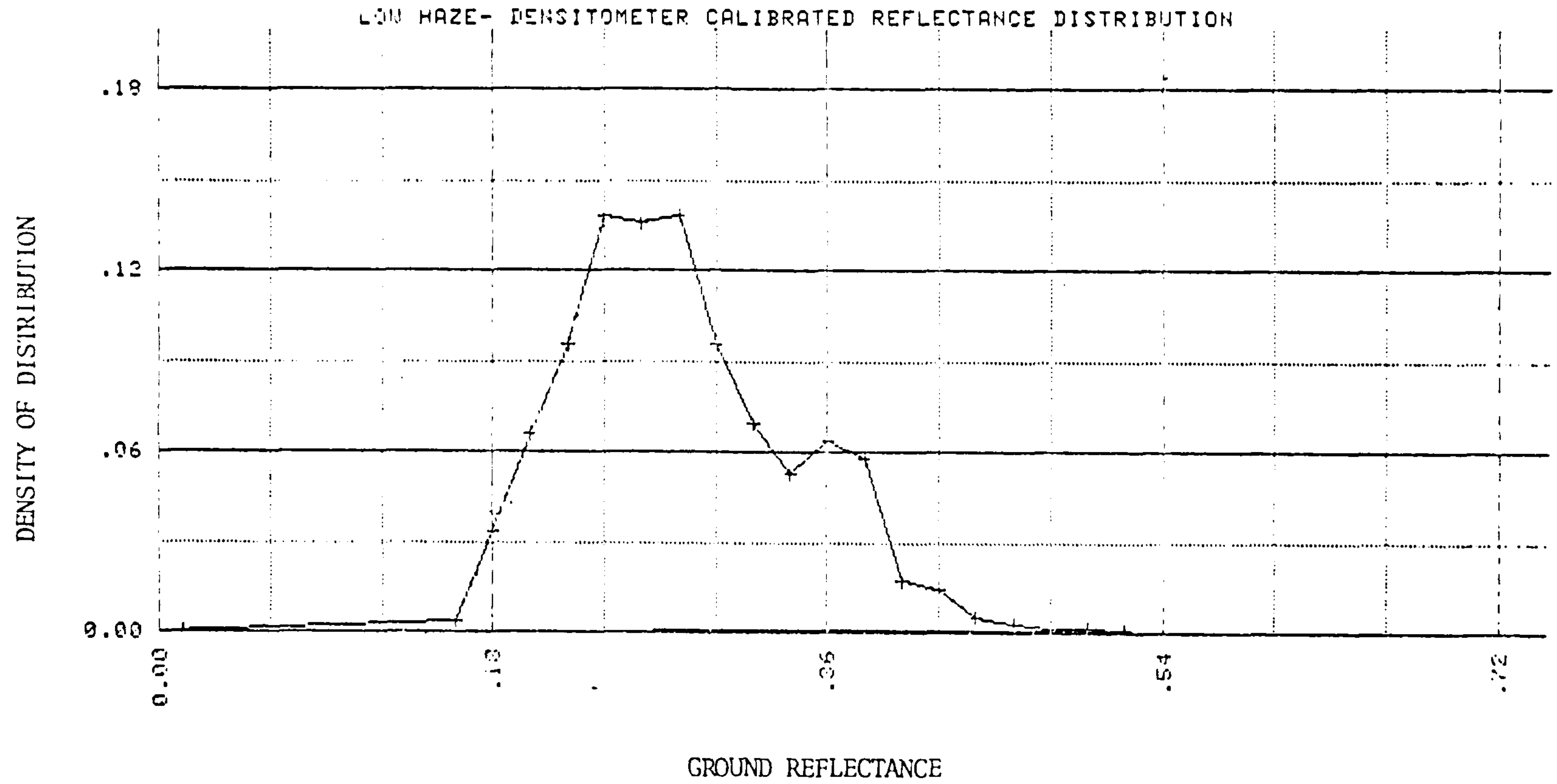


Figure 5-11 GROUND REFLECTANCE OF LOW HAZE IMAGE

for the clear picture is 0.067 and for the high haze picture is 0.097. These unequal distributions provide a good reference for comparison when evaluating the final distributions.

The second method of calibration began by selecting only six of the twelve calibration points (Figure 5-12) within the area scanned. These consisted of an area of grass, a windtunnel painted white, two areas of old blacktop, one of new blacktop, and one area of the runway. Using a reconstructed image of each scene with a computer generated retical superimposed, the coordinates of a small area enclosing each calibration point were read. Using these coordinates to retrieve the digital data, these small areas were enlarged by the computer into a character print of the area (Figures 5-13 and 5-14). Using these enlargements, a set of coordinates defining the shape and location of each calibration area was determined and used in program AREC to compute the average density of each calibration point. This is referred to as the density-averaged calibration procedure. Tables 5-1 and 5-2 show the density and the expected reflectance of each of these calibration points for the low haze and high haze scenes respectively. These tables also include calibration density values that were obtained by converting each pixel to relative exposure before averaging and then converting this exposure average back to density. This is referred to as the exposure-averaged calibration procedure.

A third set of calibration densities were used as an optimum set for comparison purpose. These consist of the density-averaged values for the low haze image (Table 5-2) and a distribution transformed set (recorded in Table 5-1) for the high haze image. This transformed set was calculated using the mean and standard deviation of each image to convert each calibration area

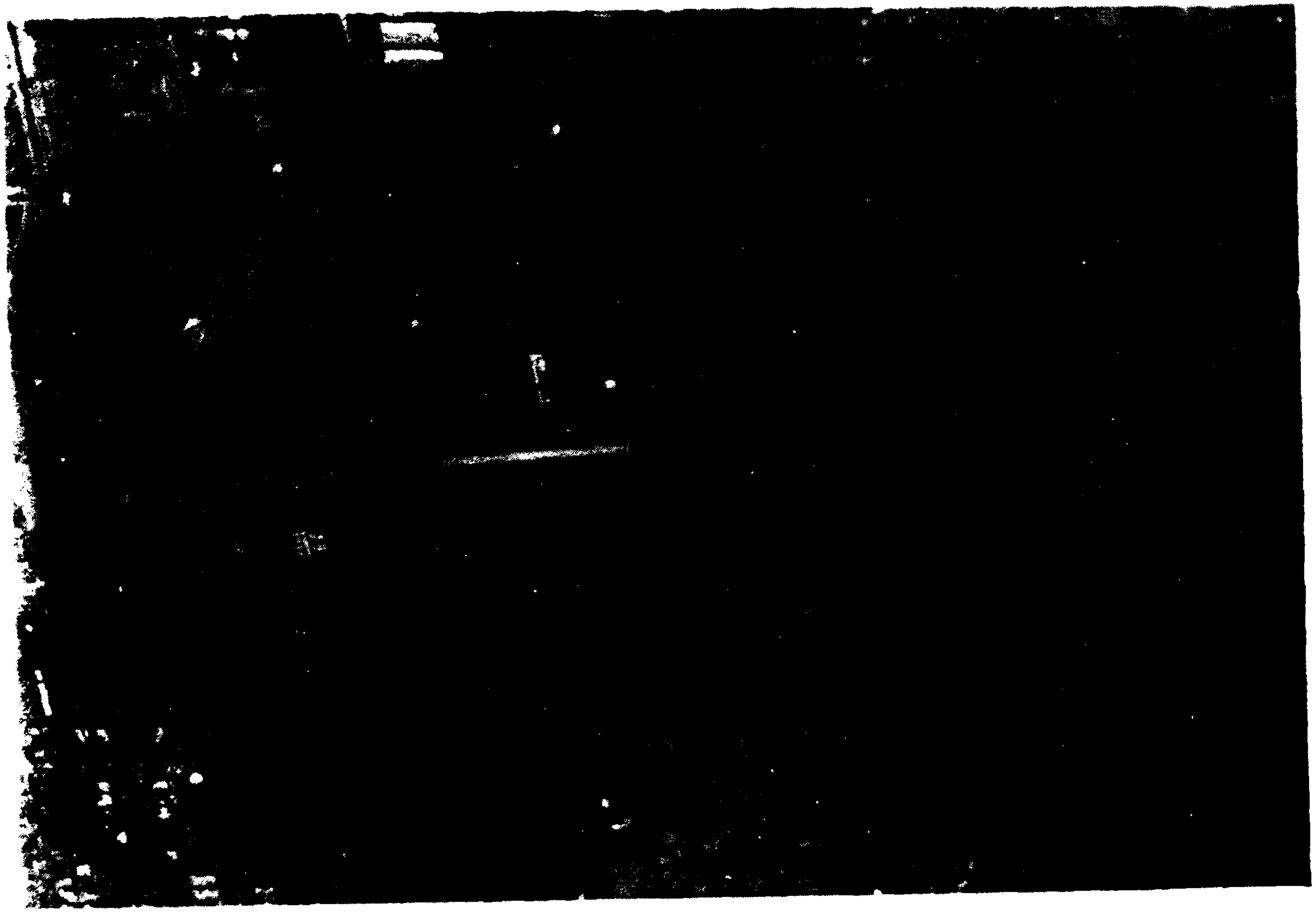


Figure 5-12 Six Objects Used as Calibration Objects

AD-A088 884 PATTERN ANALYSIS AND RECOGNITION CORP ROME NY F/G 9/2
IMAGE FEATURE MANUSCRIPT GENERATION.(U)
MAY 80 M J ZORACKI, D A MARKS, J L CAMBIER F30602-78-C-0017
UNCLASSIFIED PAR-80-25 RADC-TR-80-176 NL

2 of 2
AD
A088884

END
DATE
FILMED
10-80
DTIC

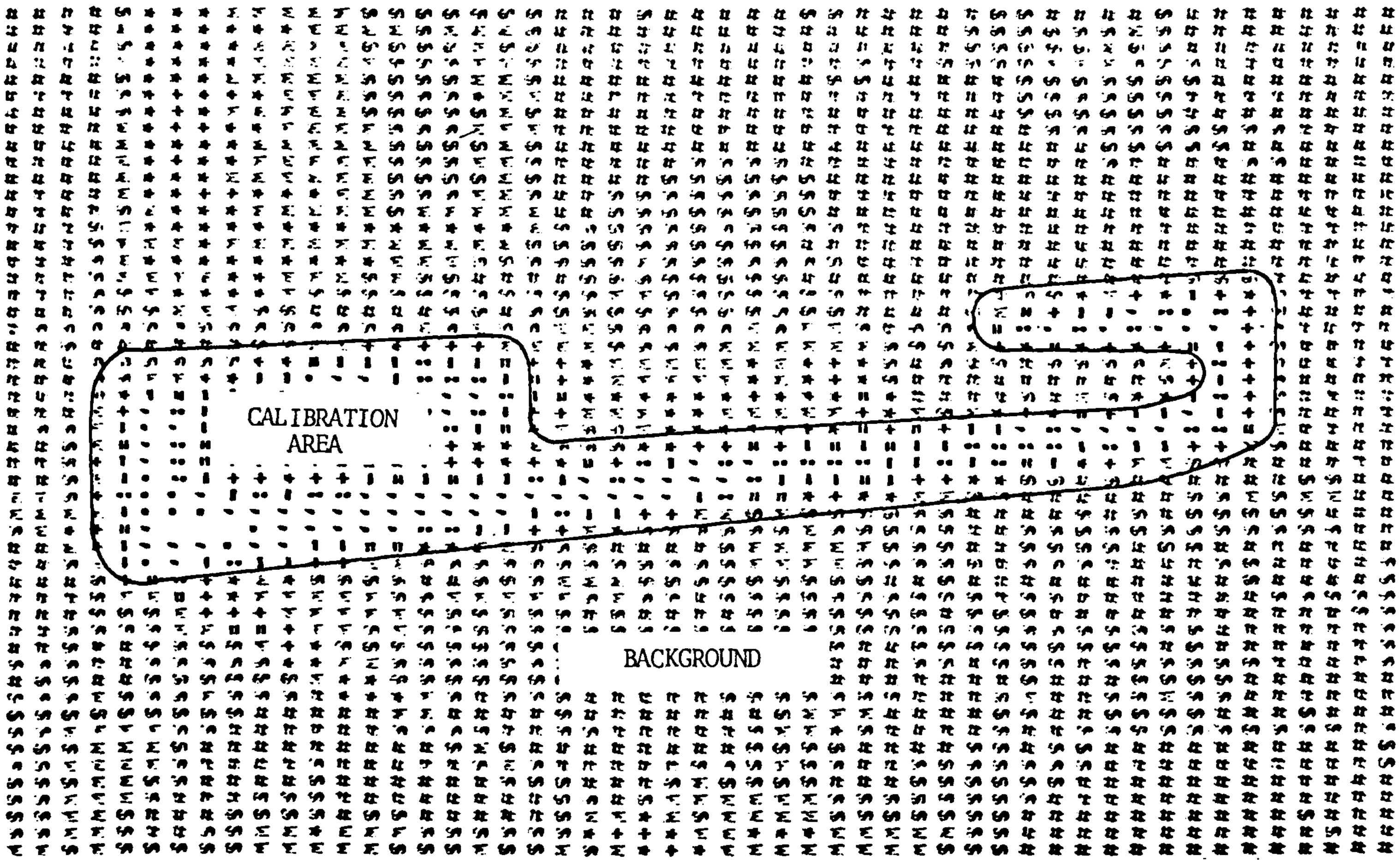


Figure 5-13 CHARACTER PICTURE OF CALIBRATION OBJECT 2 FOR HIGH HAZE SCENE

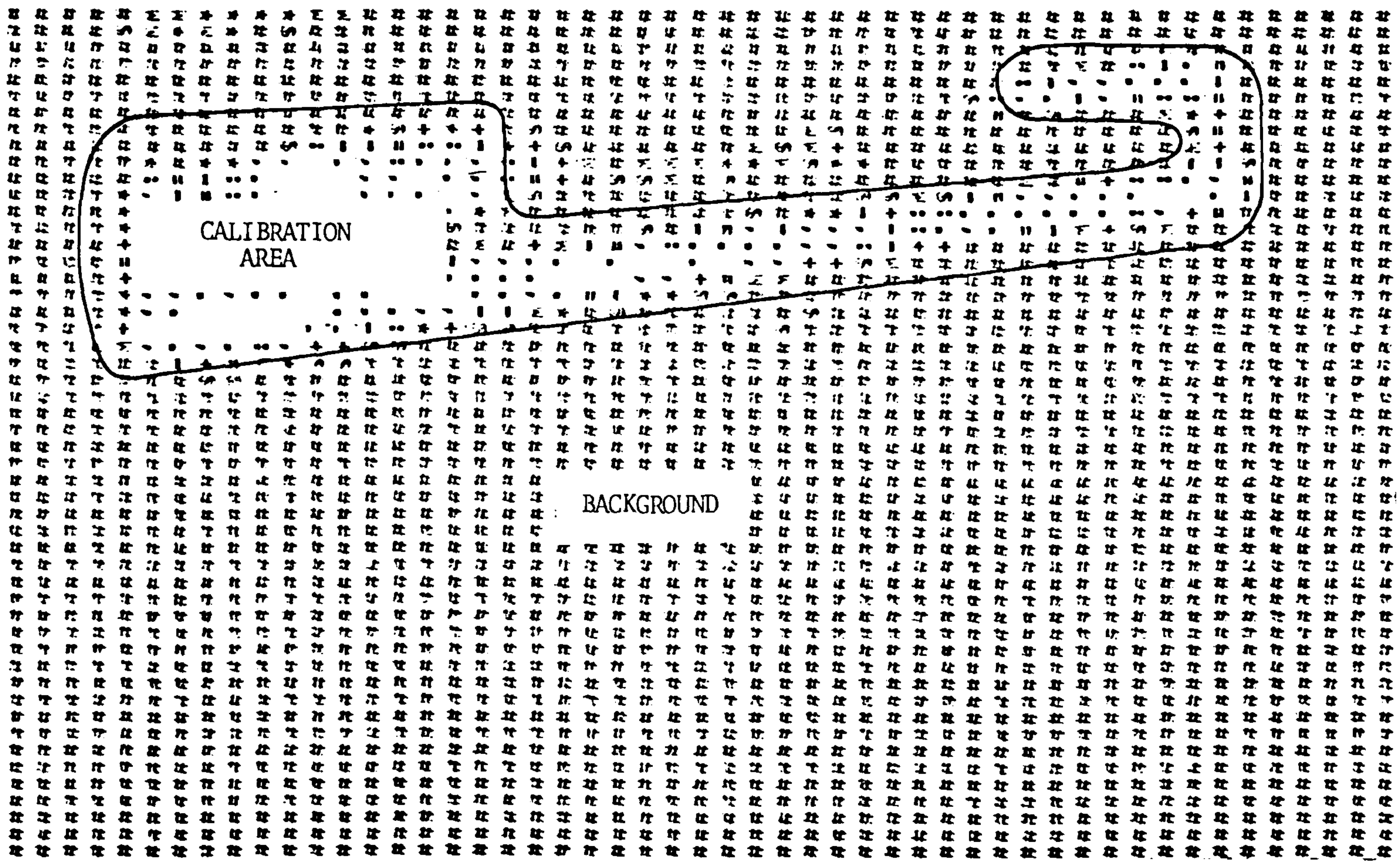


Figure 5-14 CHARACTER PICTURE OF CALIBRATION OBJECT 2 FOR LOW HAZE SCENE

Table 5-1 Calibration Areas From High Haze Scene

		Density Average	Density of Exposure Average	Expected Reflectance	Distribution Transformed Densities
1	Grass	1.899	1.905	15%	1.933
2	Wind Tunnel*	2.239	2.245	70%	2.225
5	Blacktop (old)	1.975	1.982	24%	1.988
6	Blacktop (old)	1.998	2.003	24%	1.998
7	Runway	2.077	2.078	35%	2.072
11	Blacktop (new)	1.960	1.966	10%	1.930

*White Paint

Table 5-2 Calibration Areas From Low Haze Scene

		Density Average	Density of Exposure Average	Expected Reflectance
1	Grass	1.226	1.227	15%
2	Wind Tunnel*	2.140	2.141	70%
5	Blacktop (old)	1.398	1.399	24%
6	Blacktop (old)	1.431	1.432	24%
7	Runway	1.661	1.664	35%
11	Blacktop (new)	1.217	1.219	10%

*White Paint

from the low haze image to a corresponding density in the high haze image. Using this third set of calibration values is equivalent to modifying the density values of the high haze image such that its histogram is nearly equal to that of the low haze image <u>before processing</u>. Naturally this is not for correcting the contrast of either image, but is is instead an additional means of comparing the previous corrections. This transformation was calculated and recorded, in the right hand column of Table 5-1 using

$$D_{HH} \quad (D_{LH} - \bar{D}_L)\frac{S_H}{S_L} + D_H \qquad (14)$$

where D_{HH} = the calculated distribution transformed density for high haze (Table 5-1) calibration area

D_{LH} = the calibration area density-averaged for low haze from Table 5-2

$\bar{D}_H$ = the average image density of the high haze scene

$\bar{D}_L$ = the average image density of the low haze scene

S_L = the standard deviation of the low haze image density

and S_H = the standard deviation of the high haze image density

Using the measured parameters of

$D_H = 2.006$ $\quad S_H = .0558$

and $D_L = 1.455$ $\quad S_L = .1749$

results in the equation

$$D_{HH} = (.31902)D_{LH} + 1.5418 \qquad (15)$$

By plotting the high haze density calibration values versus the corresponding low haze values and performing a linear regression, a similar equation results for each set of calibration values. These coefficients were then compared as a first evaluation of the accuracy of the restoration technique. The curve (Figure 5-15) for the calibration areas scanned separately from the image yields the equation

$$D_{HH} = (.37004)D_{LH} + 1.340 \qquad (16)$$

Using the density-averaged calibration values which were calculated from areas extracted from the scanned image, yields (Figure 5-16)

$$D_{HH} = (.33926)D_{LH} + 1.51165 \qquad (17)$$

and the same data exposure-averaged yields

$$D_{HH} = (.3380)D_{LH} + 1.5182 \qquad (18)$$

The coefficients of equations (17) and (18) indicate there is little difference between density averaging and exposure averaging. However, there is a significant difference between the coefficients of equation (17) or (18) and those of equation (16). The calibration areas corresponding to equations (17) and (18) were extracted from the digitized image directly and those corresponding to equation (16) were scanned separately from the scanning of the image. Since the coefficients of equations (17) and (18) are much closer to the distribution transformed density coefficients of equation (15), using the calibration areas of the digitized images appears to be the more reliable of the two methods. This is further demonstrated by the following analysis.

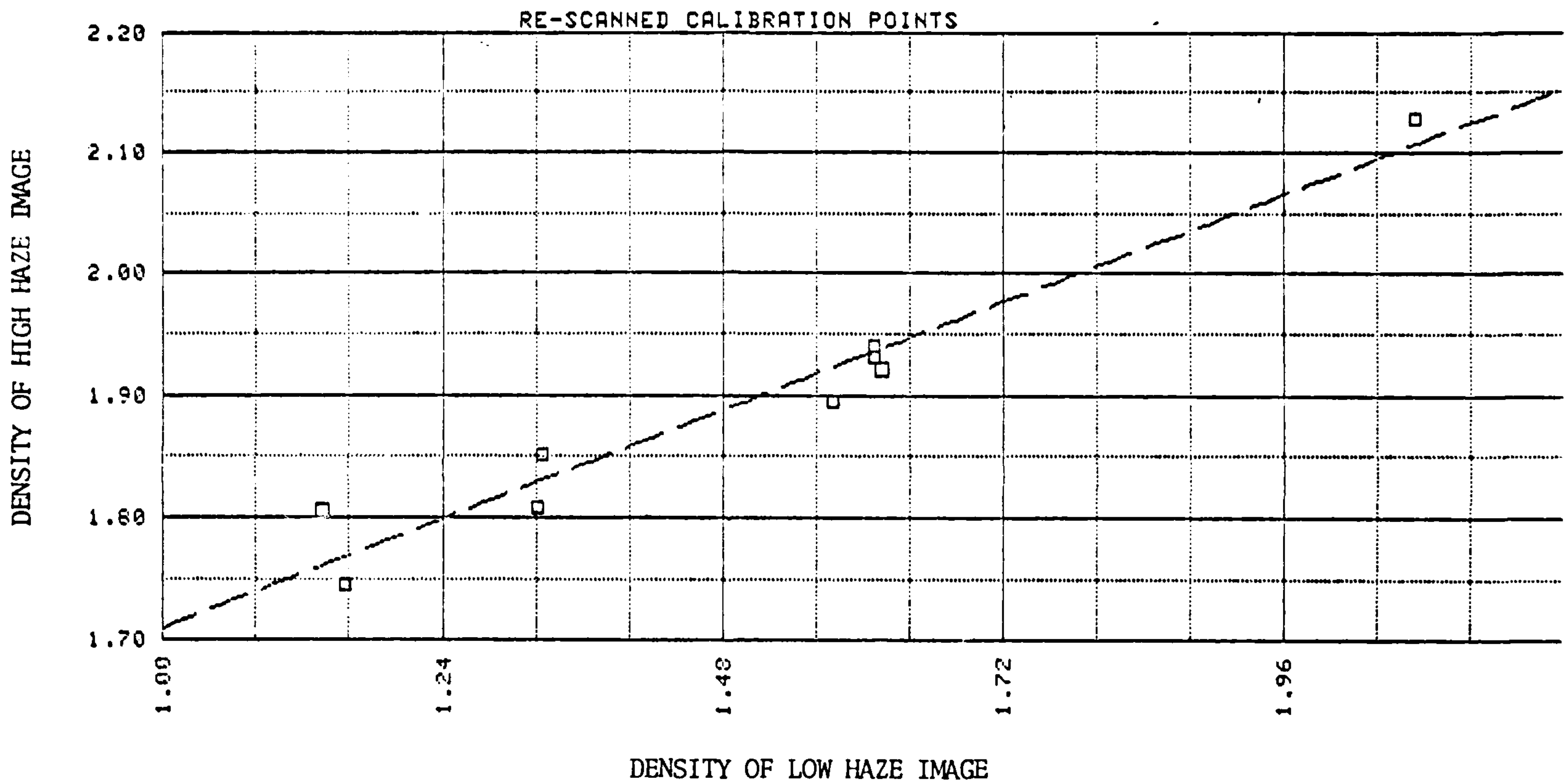


Figure 5-15 PLOT OF CALIBRATION OBJECTS SCANNED SEPARATELY FROM IMAGE

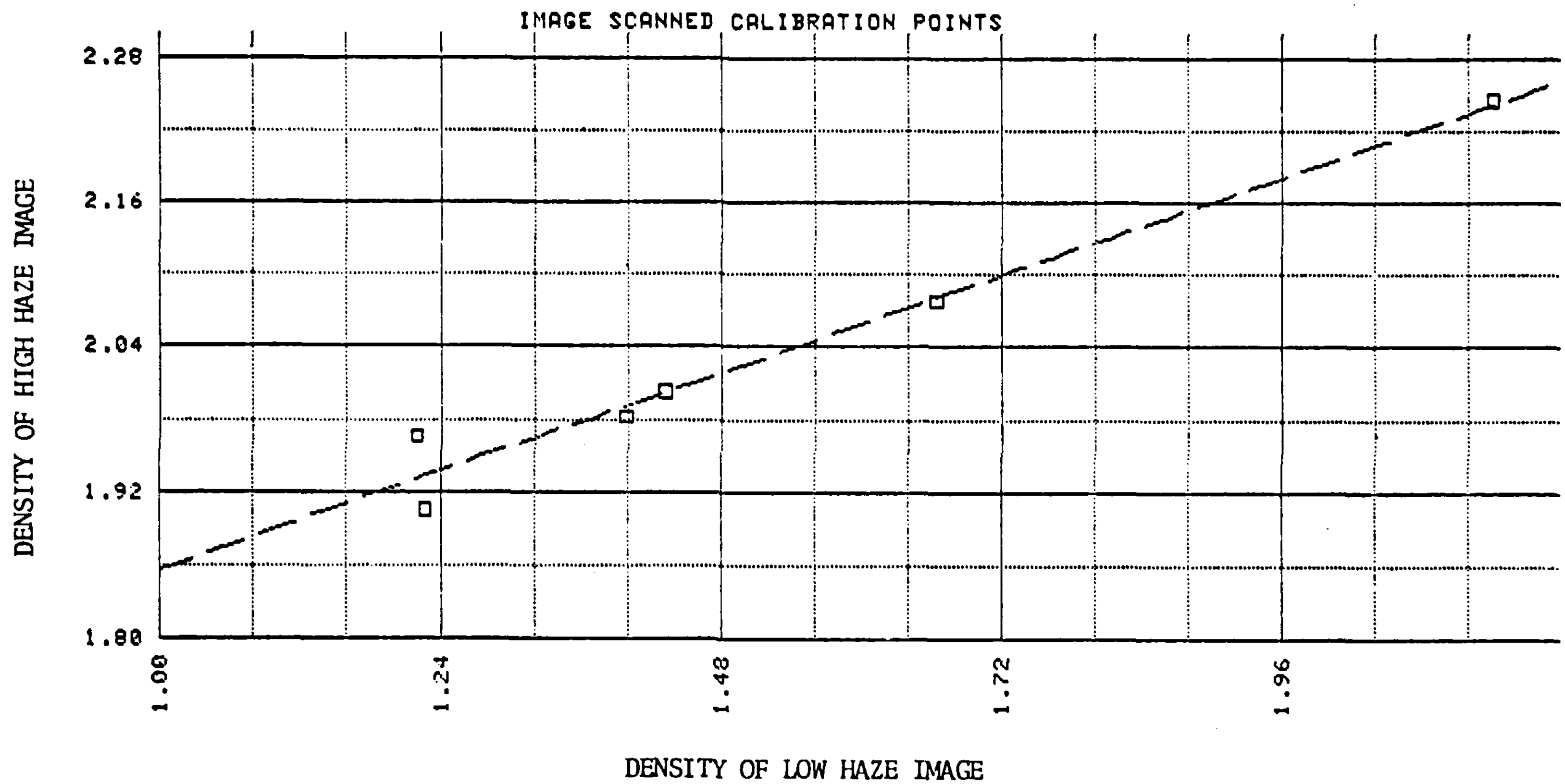


Figure 5-16 PLOT OF CALIBRATION OBJECTS TAKEN FROM DIGITIZED IMAGE

The histograms of the restored reflectance data are used as a second evaluation criterion. The restored high haze image and the restored low haze image should transform into two images having nearly equal histograms. The density-averaged, the exposure-averaged, and the histogram- corrected calibration sets all give similar histograms (Figures 5-17, 5-18, 5-19, 5-20, and 5-21 respectively). However, using the separately scanned areas for calibration yielded restored images with histograms (Figures 5-7 and 5-8) that were inconsistent with the other sets of calibration data. These two histograms were also not similar to each other since the mean of the high haze distribution was much higher than the low haze distribution.

Of course, the final and most convincing comparison is the appearance of the resulting imagery. The seven sets of calibration area densities, together with their corresponding image data and D Log E curve, were processed by the program AREFL. This program calculates a table of 301 reflectance values indexed by a corresponding density value in the range of 0.00 to 3.00 density units for the particular image. This table is used with the image data tapes during processing as a look up table to generate the final image reflectance tapes. This table, for each of the seven sets of calibration data, is plotted in Figures 5-22 through 5-28. The reconstructed images from the reflectance tapes are shown in Figures 5-29, 5-30, and 5-31. Comparing these with the original photographs (Figures 5-2 and 5-3) the successful contrast restoration of the aerial imagery is obvious. However, there is a slight variation in picture brightness from left to lower right in the high haze picture which indicates that there was heavier haze on the lower right side. However, the

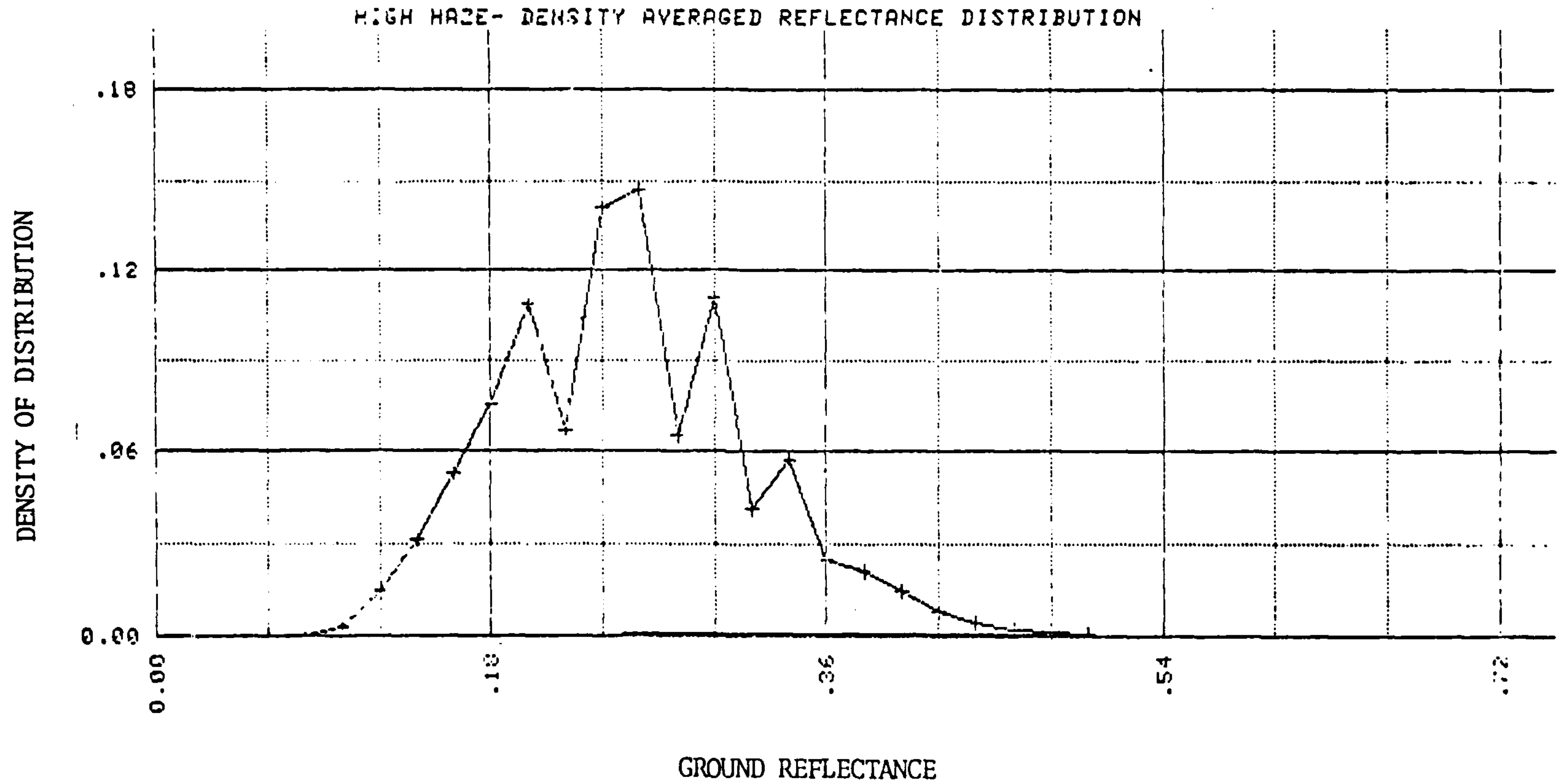


Figure 5-17 HISTOGRAM OF GROUND REFLECTANCE USING DIGITIZED IMAGE CALIBRATION AREAS (HIGH HAZE, DENSITY-AVERAGED)

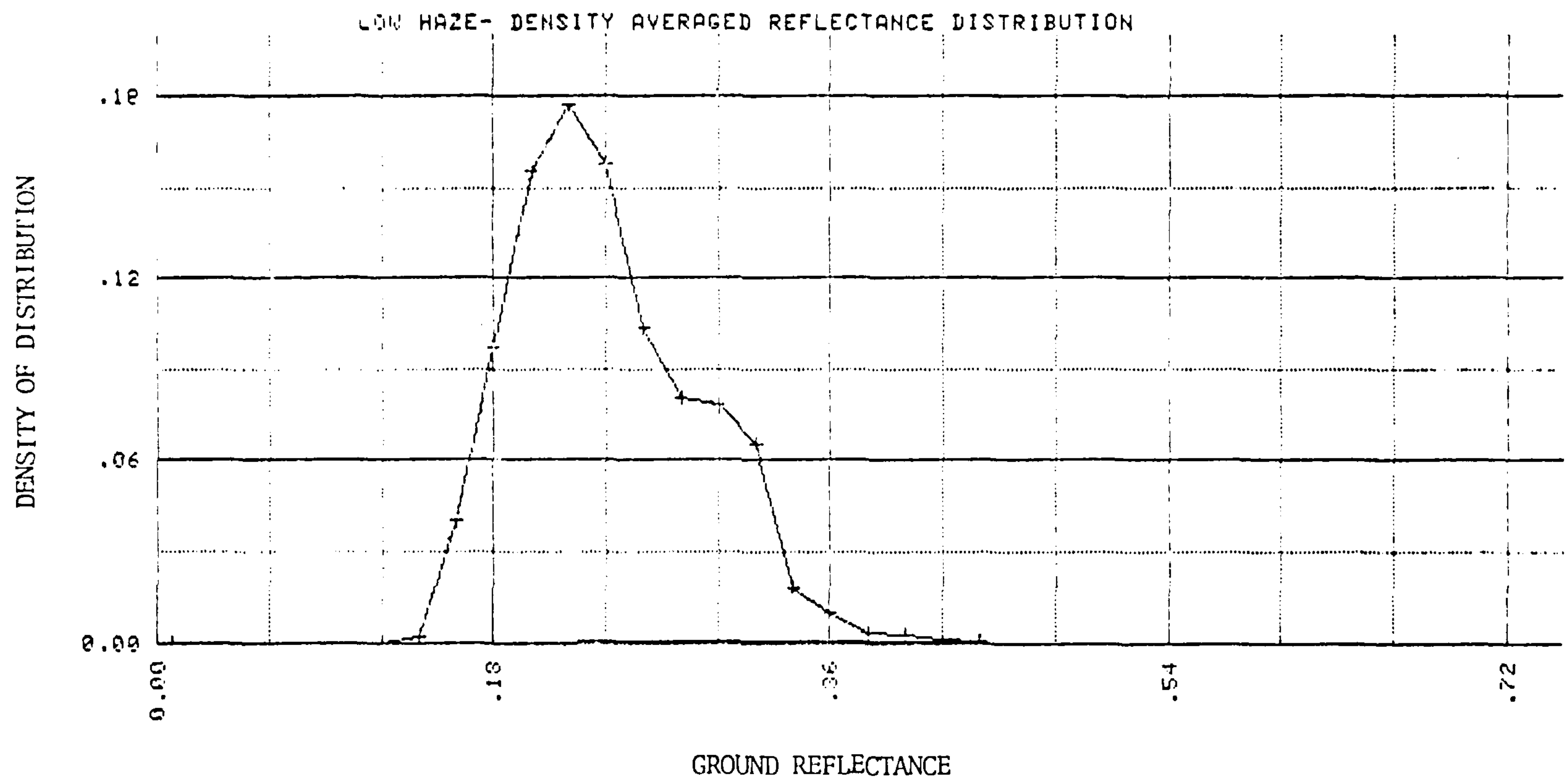


Figure 5-18 HISTOGRAM OF GROUND REFLECTANCE USING DIGITIZED IMAGE CALIBRATION AREAS (LOW HAZE, DENSITY-AVERAGED)

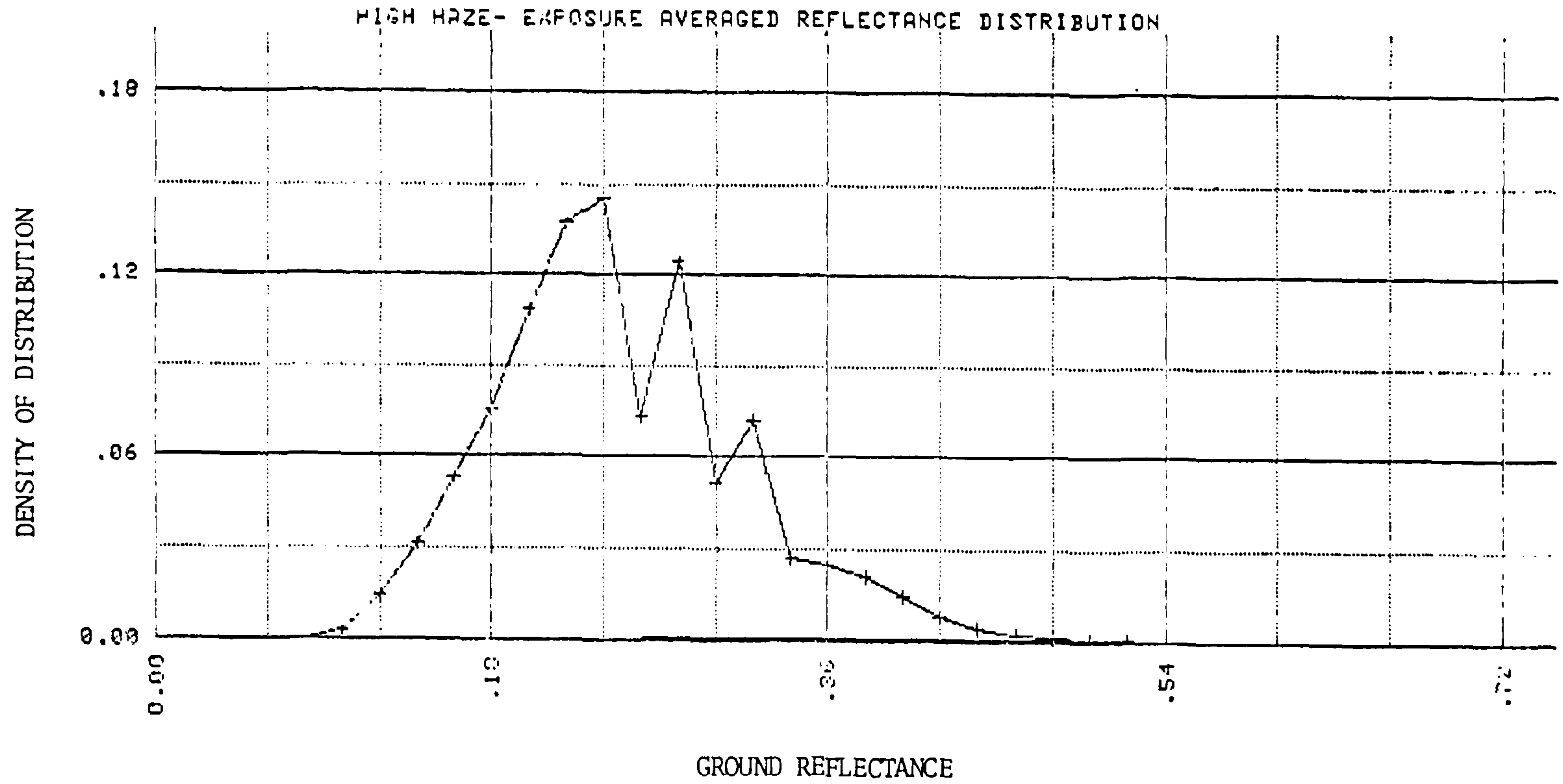


Figure 5-19 HISTOGRAM OF GROUND REFLECTANCE USING DIGITIZED IMAGE CALIBRATION AREAS (HIGH HAZE, EXPOSURE-AVERAGED)

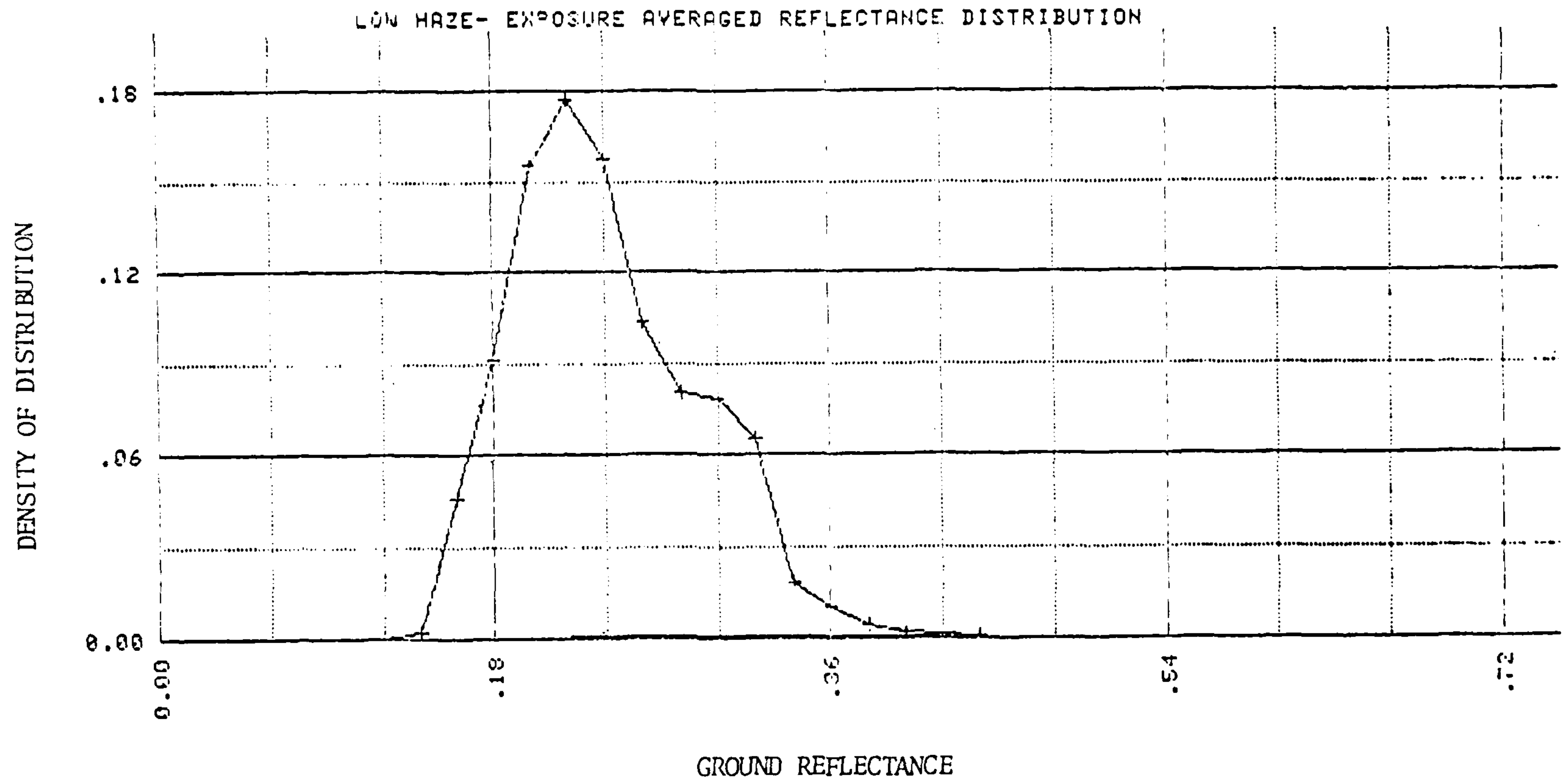


Figure 5-20 HISTOGRAM OF GROUND REFLECTANCE USING DIGITIZED IMAGE CALIBRATION AREAS (LOW HAZE, EXPOSURE-AVERAGED)

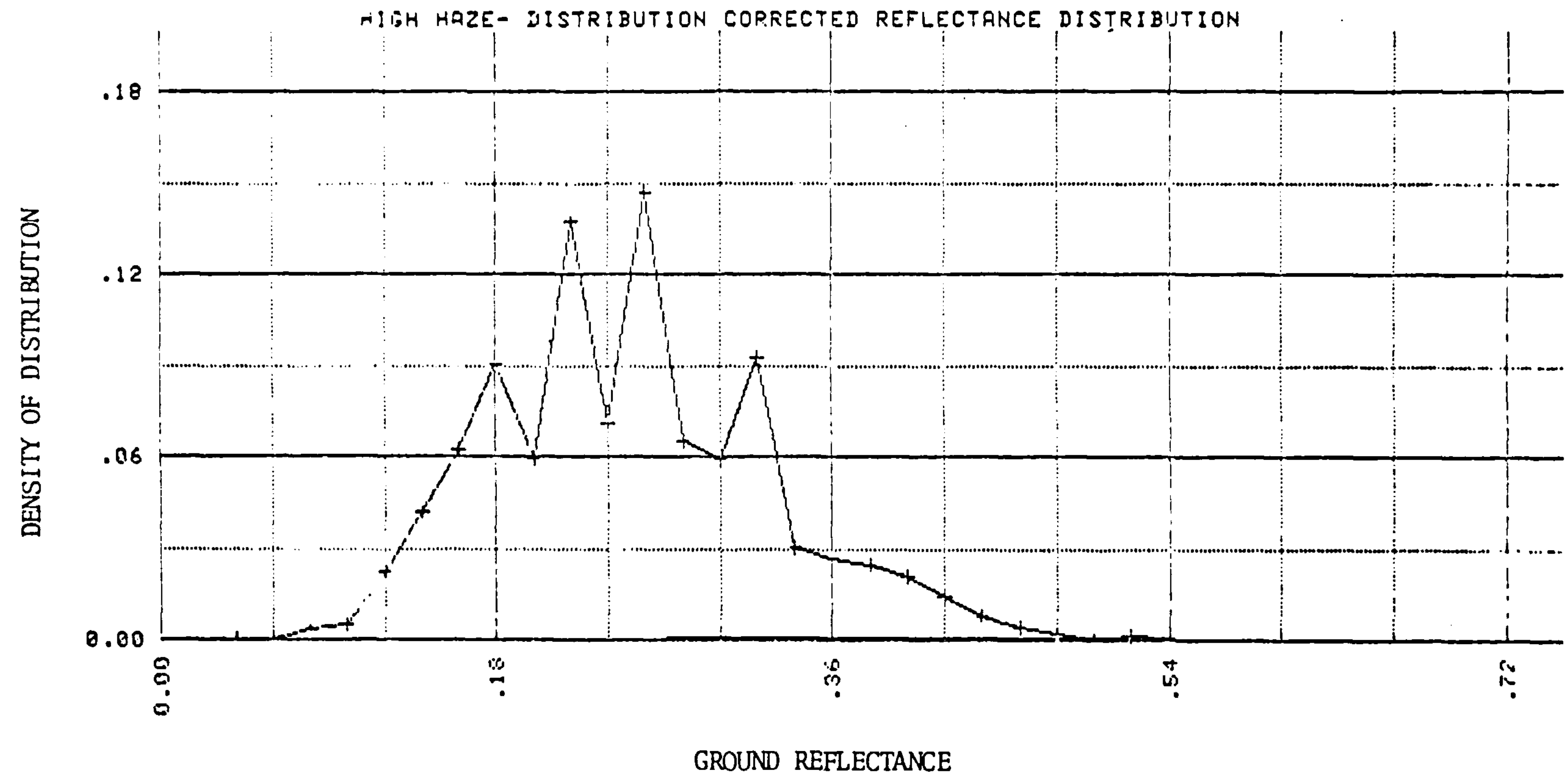


Figure 5-21 HISTOGRAM OF GROUND REFLECTANCE USING DIGITIZED IMAGE CALIBRATION AREAS (HIGH HAZE, DISTRIBUTION CORRECTED)

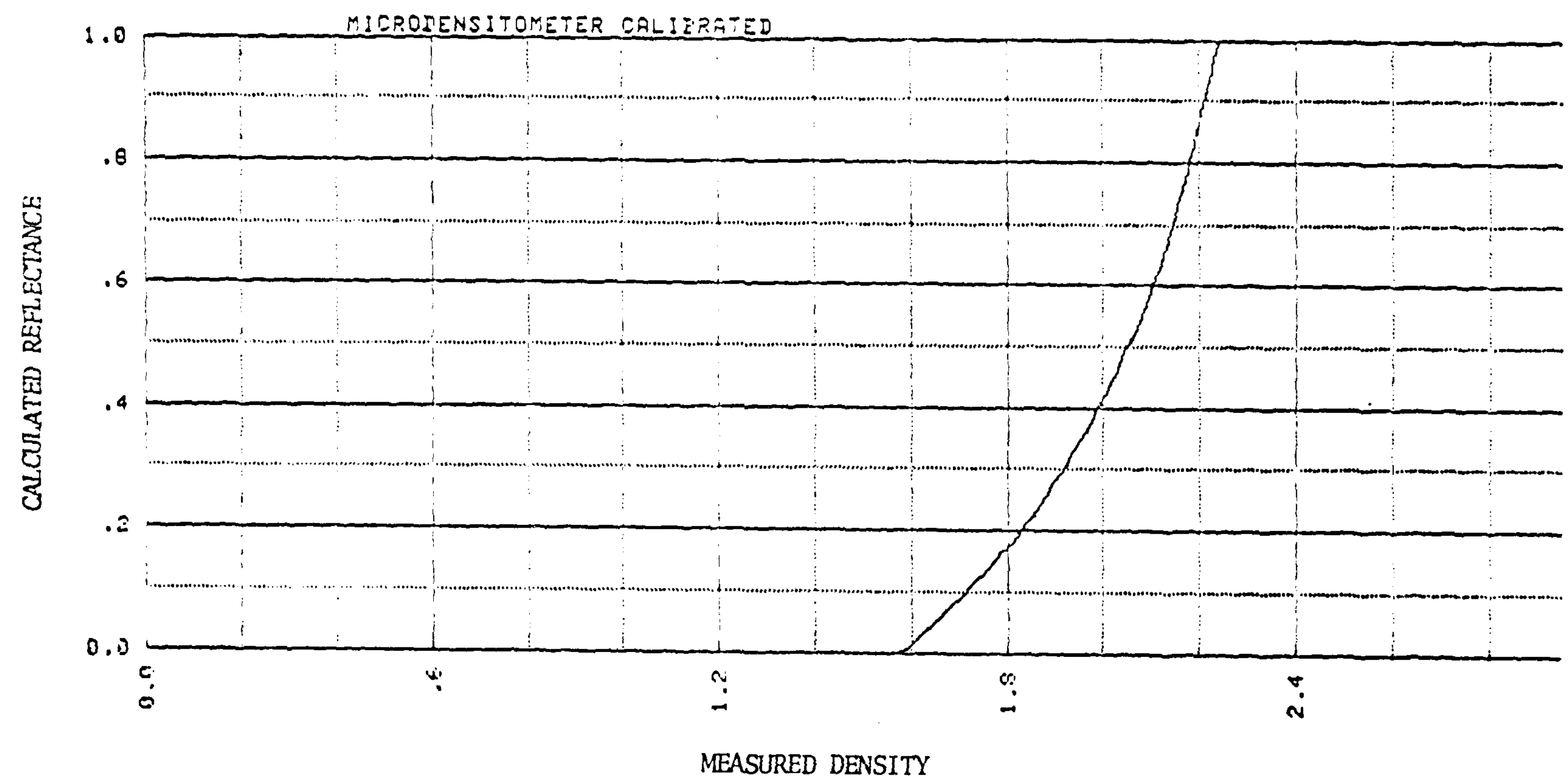


Figure 5-22 DENSITY TO REFLECTANCE CONVERSION FOR SECONDARY SCANNING OF CALIBRATION AREAS (HIGH HAZE)

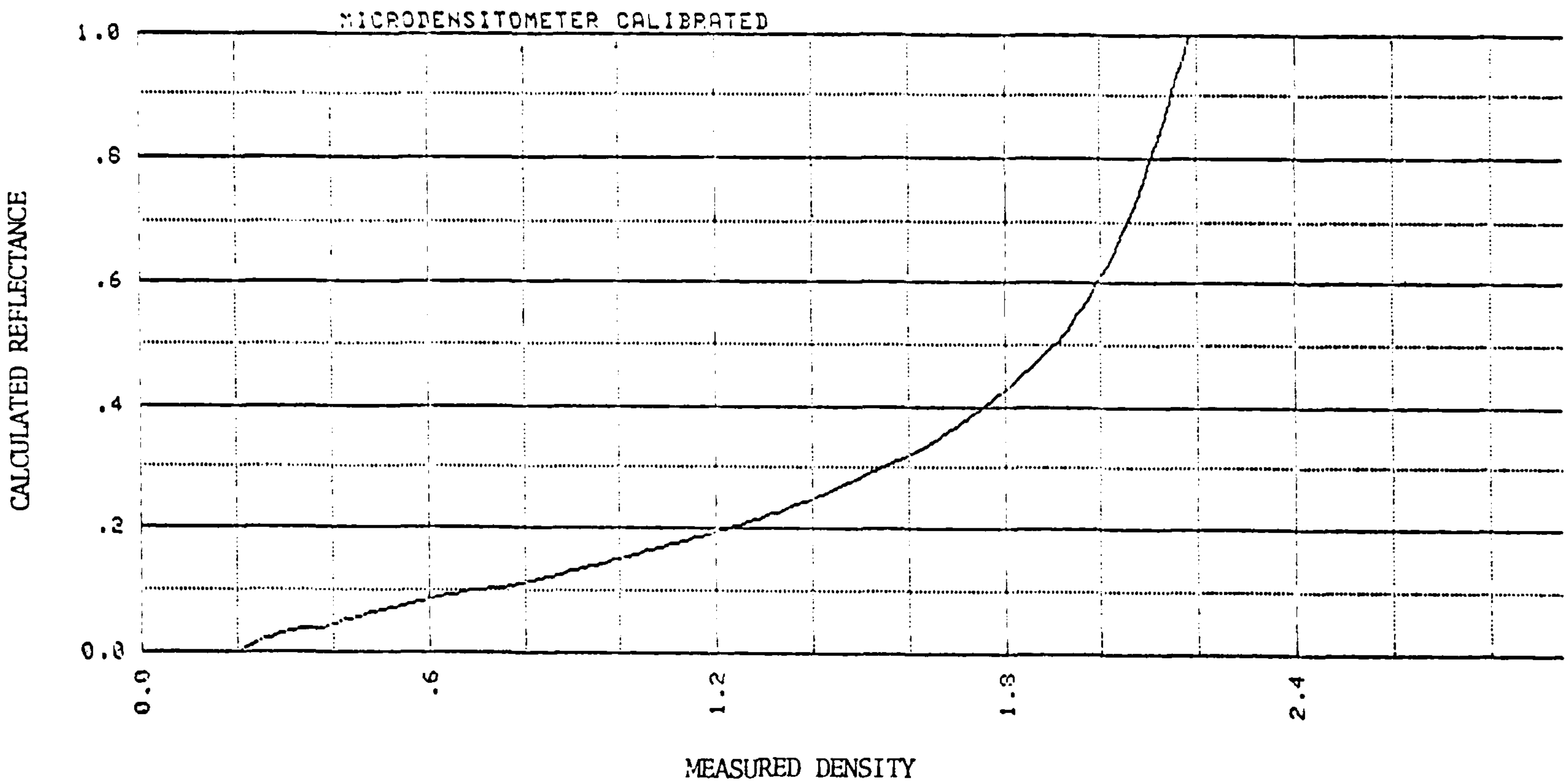


Figure 5-23 DENSITY TO REFLECTANCE CONVERSION FOR SECONDARY SCANNING OF CALIBRATION AREAS (LOW HAZE)

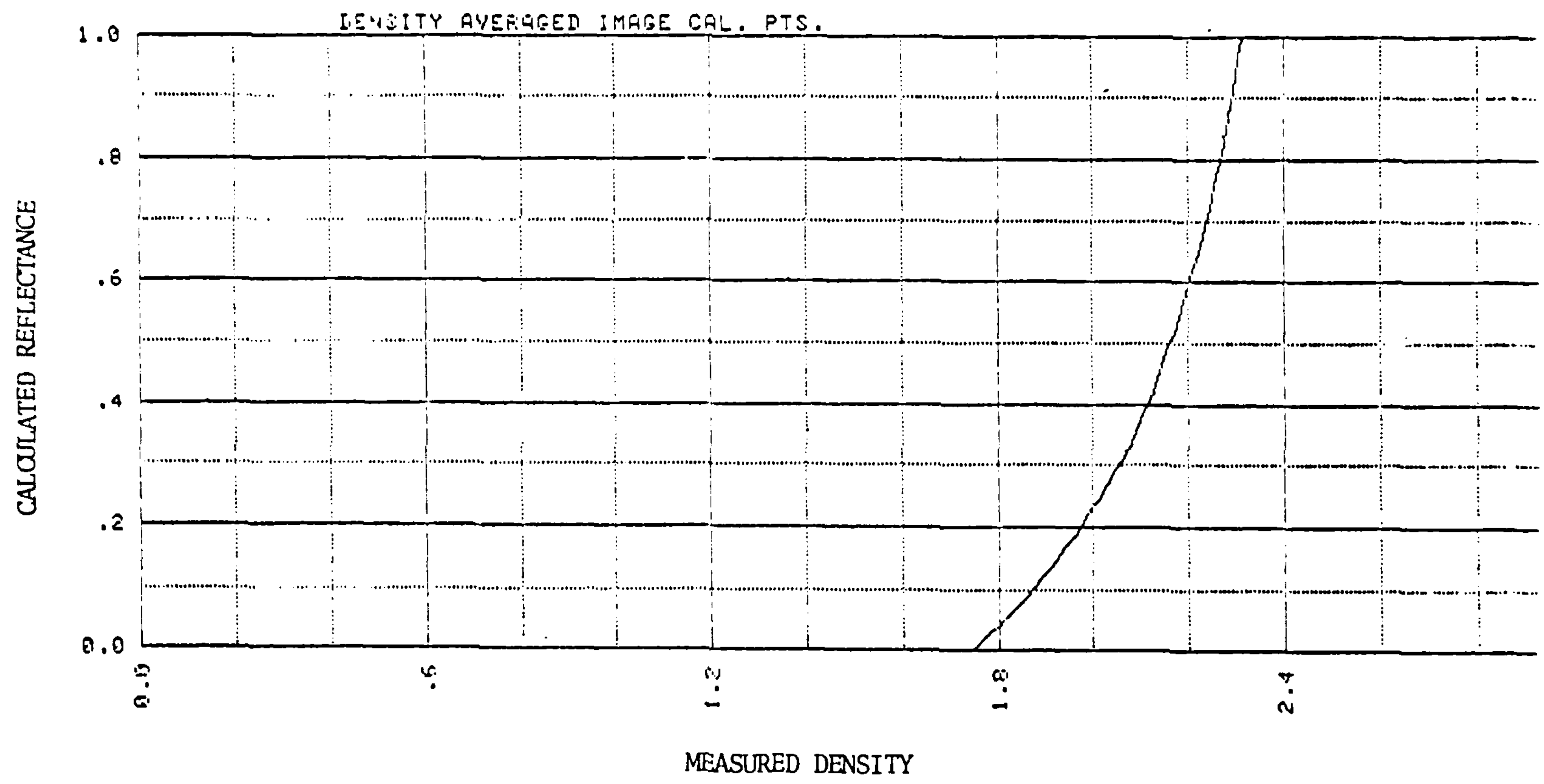


Figure 5-24 DENSITY TO REFLECTANCE CONVERSION FOR DIGITAL IMAGE CALIBRATION AREAS (HIGH HAZE, DENSITY-AVERAGED)

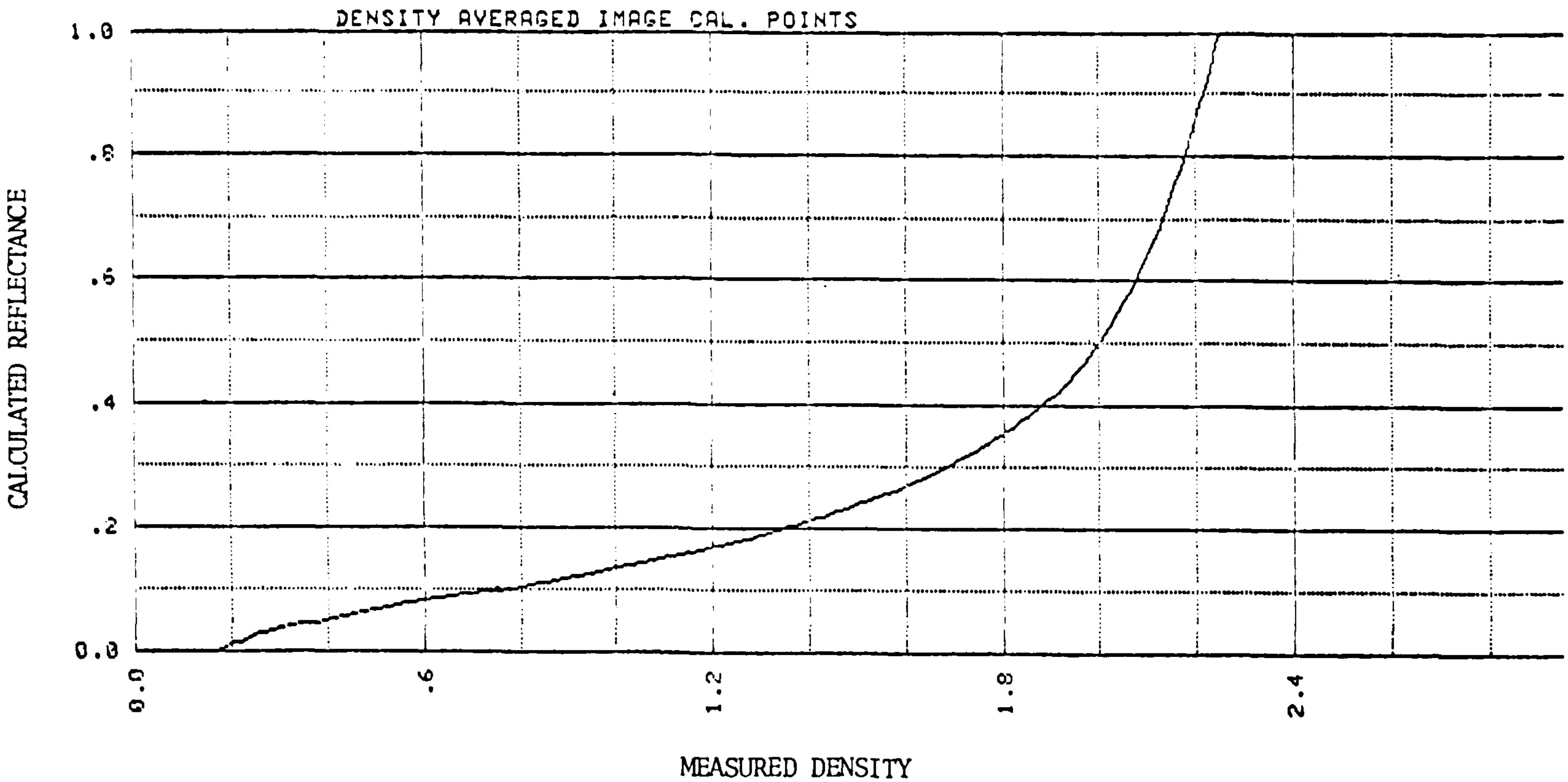


Figure 5-25 DENSITY TO REFLECTANCE CONVERSION FOR DIGITAL IMAGE CALIBRATION AREAS (LOW HAZE, DENSITY-AVERAGED)

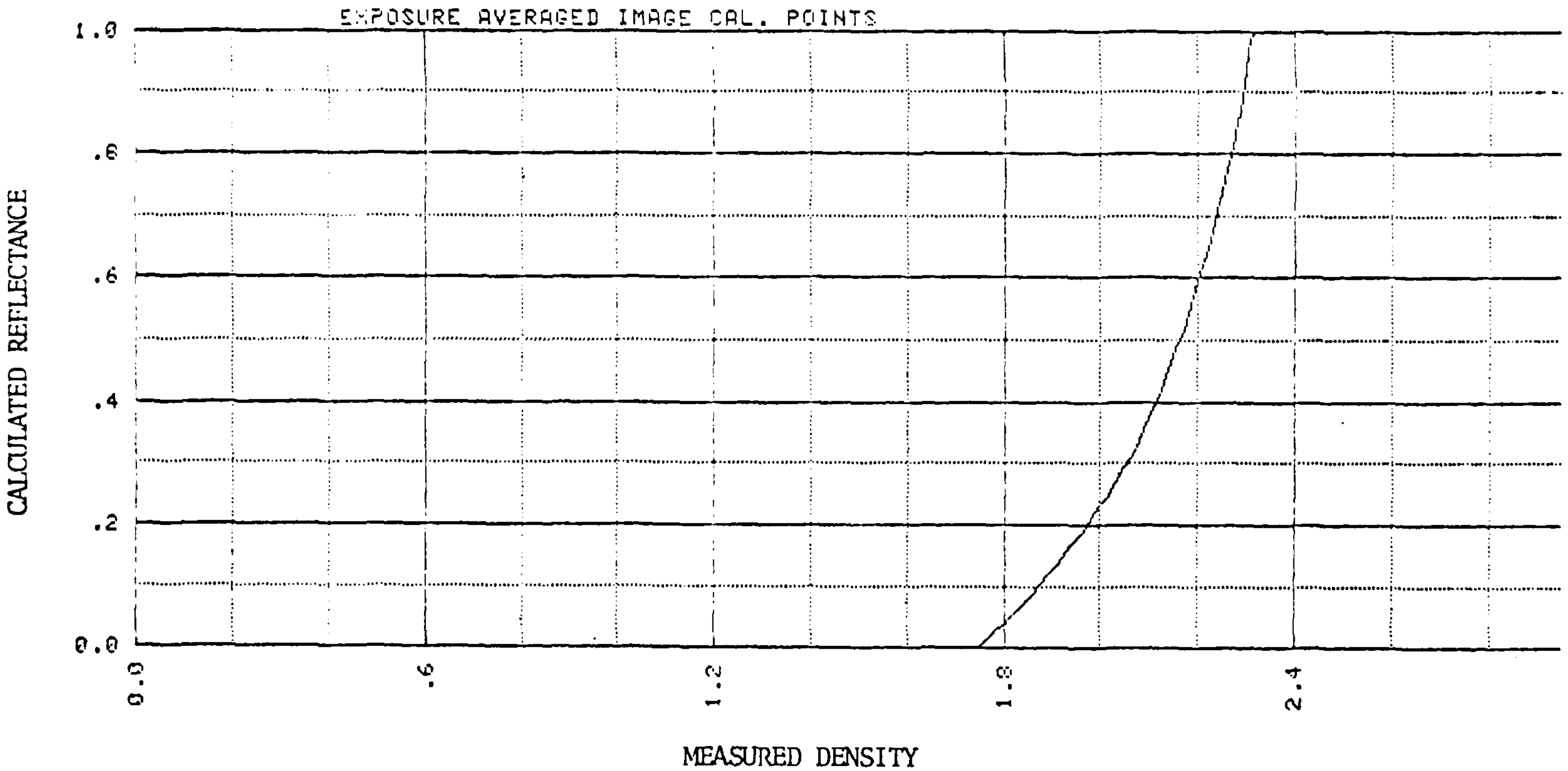


Figure 5-26 DENSITY TO REFLECTANCE CONVERSION FOR DIGITAL IMAGE CALIBRATION AREAS (HIGH HAZE, EXPOSURE-AVERAGED)

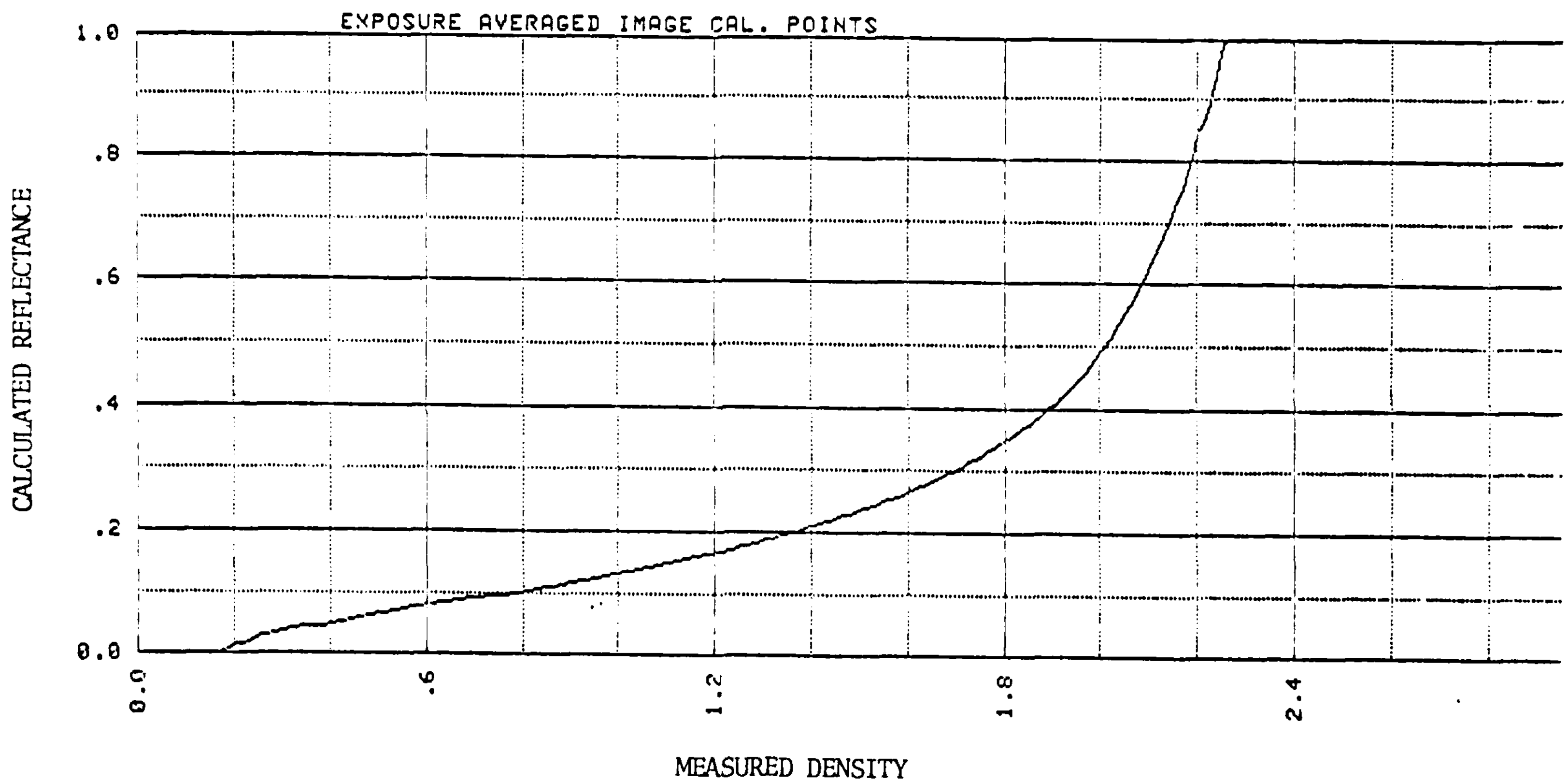


Figure 5-27 DENSITY TO REFLECTANCE CONVERSION FOR DIGITAL IMAGE CALIBRATION AREAS (LOW HAZE, EXPOSURE-AVERAGED)

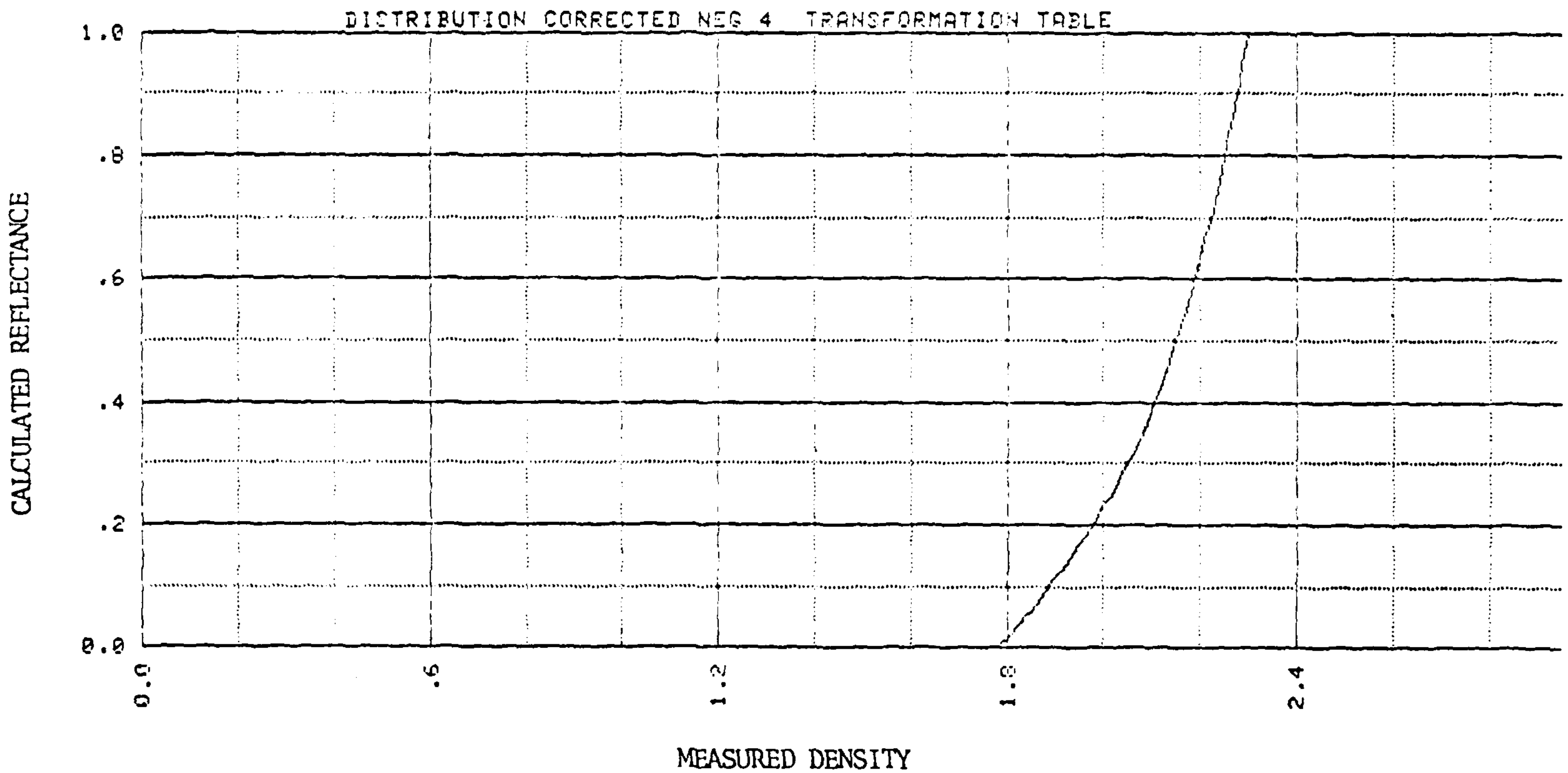
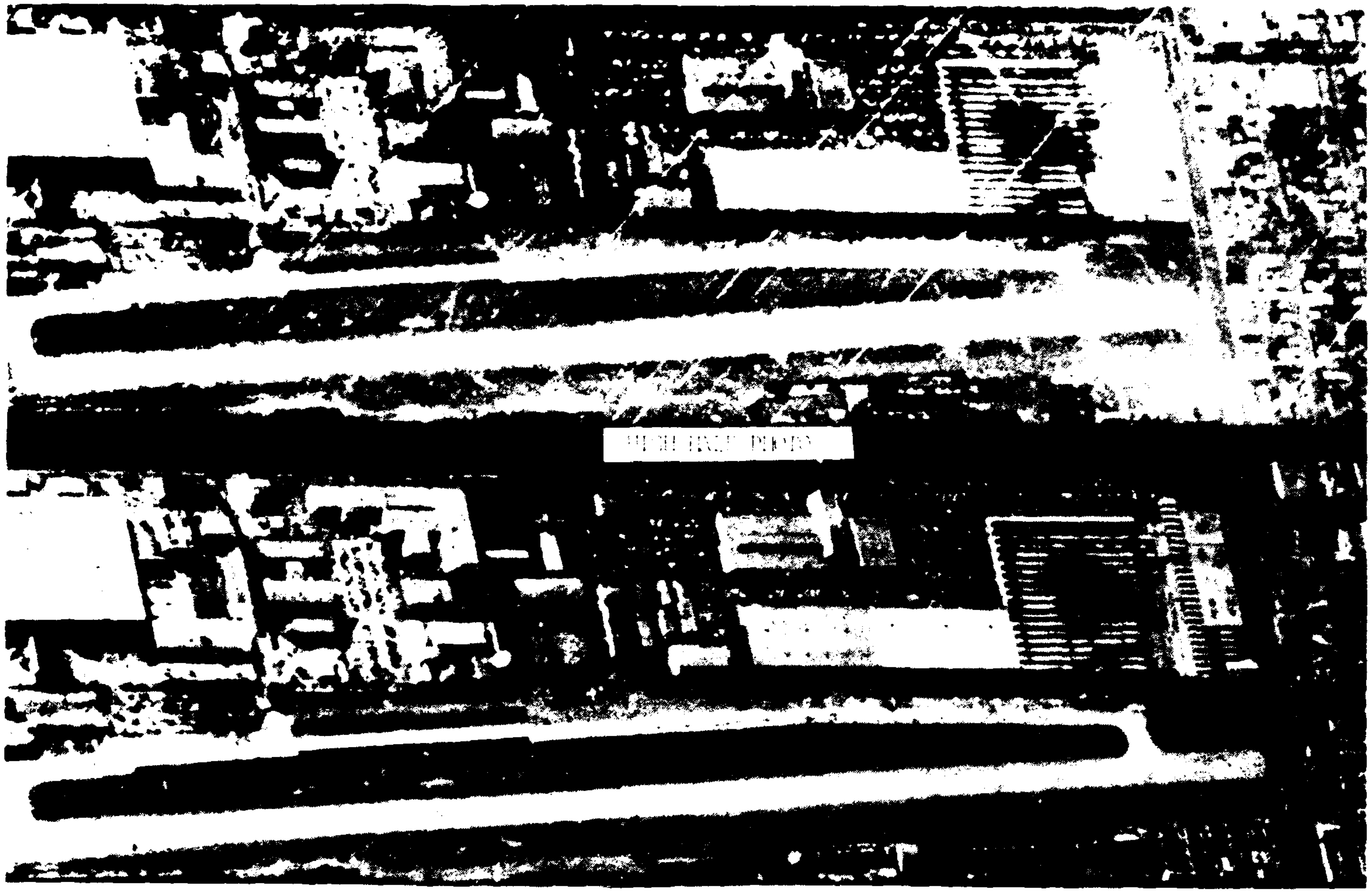


Figure 5-28 DENSITY TO REFLECTANCE CONVERSION FOR DISTRIBUTION CORRECTION CALIBRATION (HIGH HAZE)



LOW HAZE PHOTO

Figure 5-29 Restoration Results Using Density Averaging

LOW HAZE PHOTO

Figure 5-30 Restoration Results Using Exposure Averaging

LOW HAZE PHOTO

Figure 5-31 Restoration Results Using Distribution Correction on High Haze Image

overall equality in scene brightness between the two scenes of the density-averaged and the exposure-averaged pictures demonstrates the effectiveness of the contrast restoration technique.

5.5 CONCLUSIONS

This work was based upon the hypothesis that use of haze model equations could provide a technique to restore the contrast to imagery degraded by the atmospheric haze. The results of tests of the technique have indicated the following conclusions.

The contrast restoration technique based on reconstruction of ground reflectance was successful. The results were demonstrated mathematically as well as through the visual appearance of the photographic image.

Inspection of the finished photograph indicated removal of the haze effects; however, one area of the restored image still shows evidence of a small cloud.

Care is needed during the procedure in ditigizing the image to be restored. The entire image should be digitized in one step and then, from this digital data, areas of known reflectance should be selected for use in the procedure. Scanning these areas at a different time than the image resulted in significant errors.

The simplification of the equations, resulting in the use of look up tables for the procedure, drastically reduced the complexity of the computations and computer run time requirements.

The contrast restoration technique improves the appearance of a photograph, however this technique did not address restoring image edge sharpness and overall resolution. This was beyond the scope of the work effort; however, the techniques do exist that can be applied for these other areas of improvement.

5.6 RECOMMENDATIONS

Successful conclusion of the work effort provided a software package for installation in the RADC Automatic Feature Extraction System. As an aid toward effective application of the contrast restoration technique, it is recommended that:

1. The contrast restoration software be installed in the Automatic Feature Extraction System.

2. The utilization effectiveness of the contrast restoration technique be studied and product improvement areas be identified and implemented.

3. The contrast restoration technique be considered for other applications where rapid transformation of image contrast and/or scene reflectance is needed.

APPENDIX A

SUMMARY OF ATMOSPHERIC EFFECTS AND "C" FACTOR ON AERIAL CONTRAST

A.1 ATMOSPHERIC FACTORS

The effects of the atmosphere and the illumination of the scene by the sun, as viewed by an observer or a sensor lens, are very complex and ever changing with time. However, with the advent of a digital computer, practical solutions to the very complex atmospheric equations now exist as models, and one can routinely examine the effects of each major atmospheric parameter upon the available energy and scene contrast. Referring to Figures A-1 and A-2, one can visualize the solar energy at the top of the atmosphere. The energy enters the multilayered or variable density atmosphere containing the molecules of air and particles of dust (defined as haze) and a portion is multipath scattered in all directions as non-image forming light while another portion of the energy is absorbed in the atmosphere, and the remainder of the light reaches the ground to illuminate the scene. The ground scene absorbs some of the energy and reflects the remainder in accordance with the reflectance characteristic of the specific material of the scene. The reflected light is transmitted through the atmosphere (again undergoing multipath scatter) to the sensor as image forming light. The haze and image forming energy are less than 100% of the available solar illumination, therefore, path transmittance losses account for most of the energy loss due to the atmosphere. The haze and image forming (scene) energy values vary as a function of wavelength as well as atmospheric and illumination parameters and can be measured or computed in watts/steradian/cm^2/cm wavelength. Path

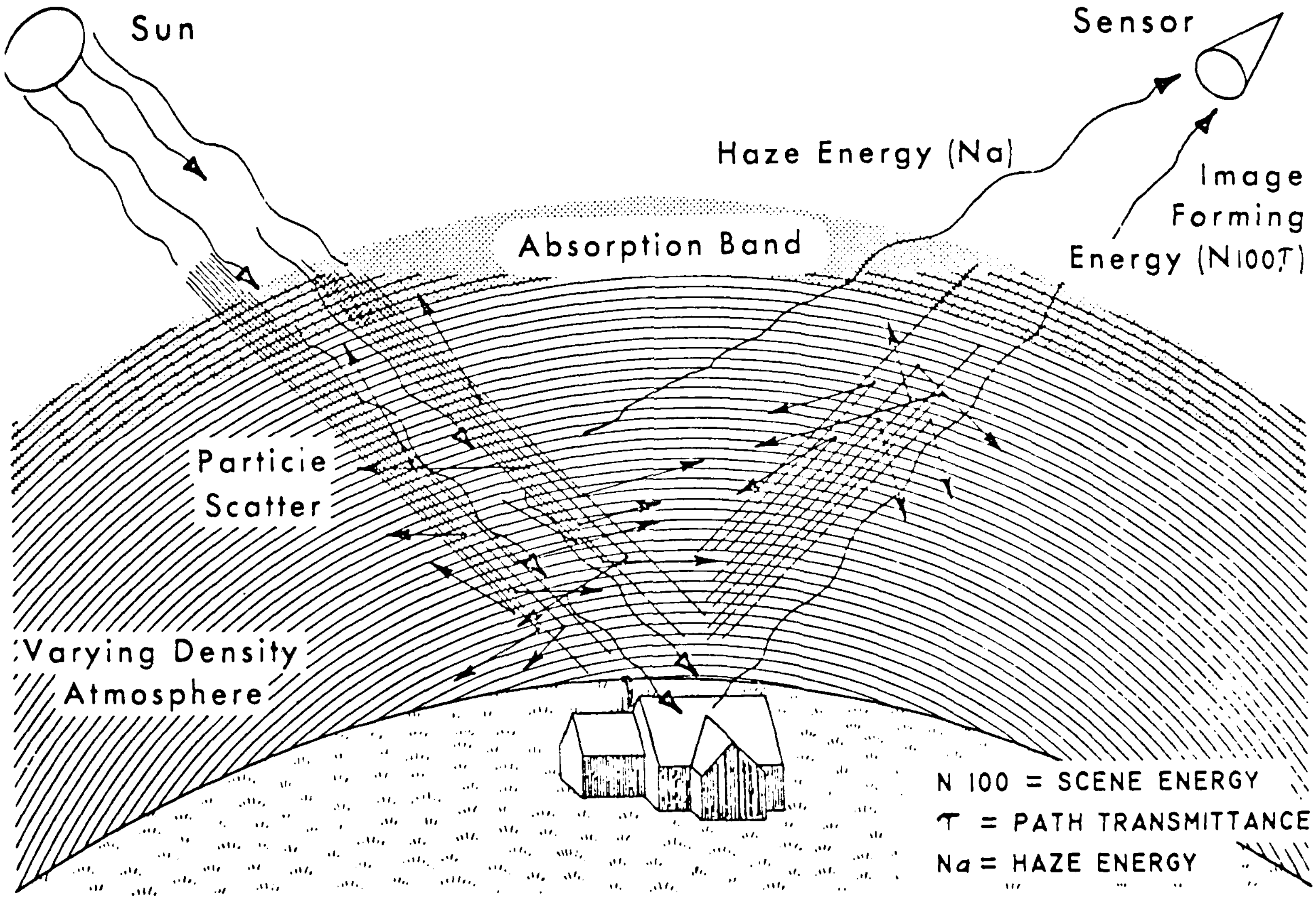


Figure A-1 Atmospheric Factors

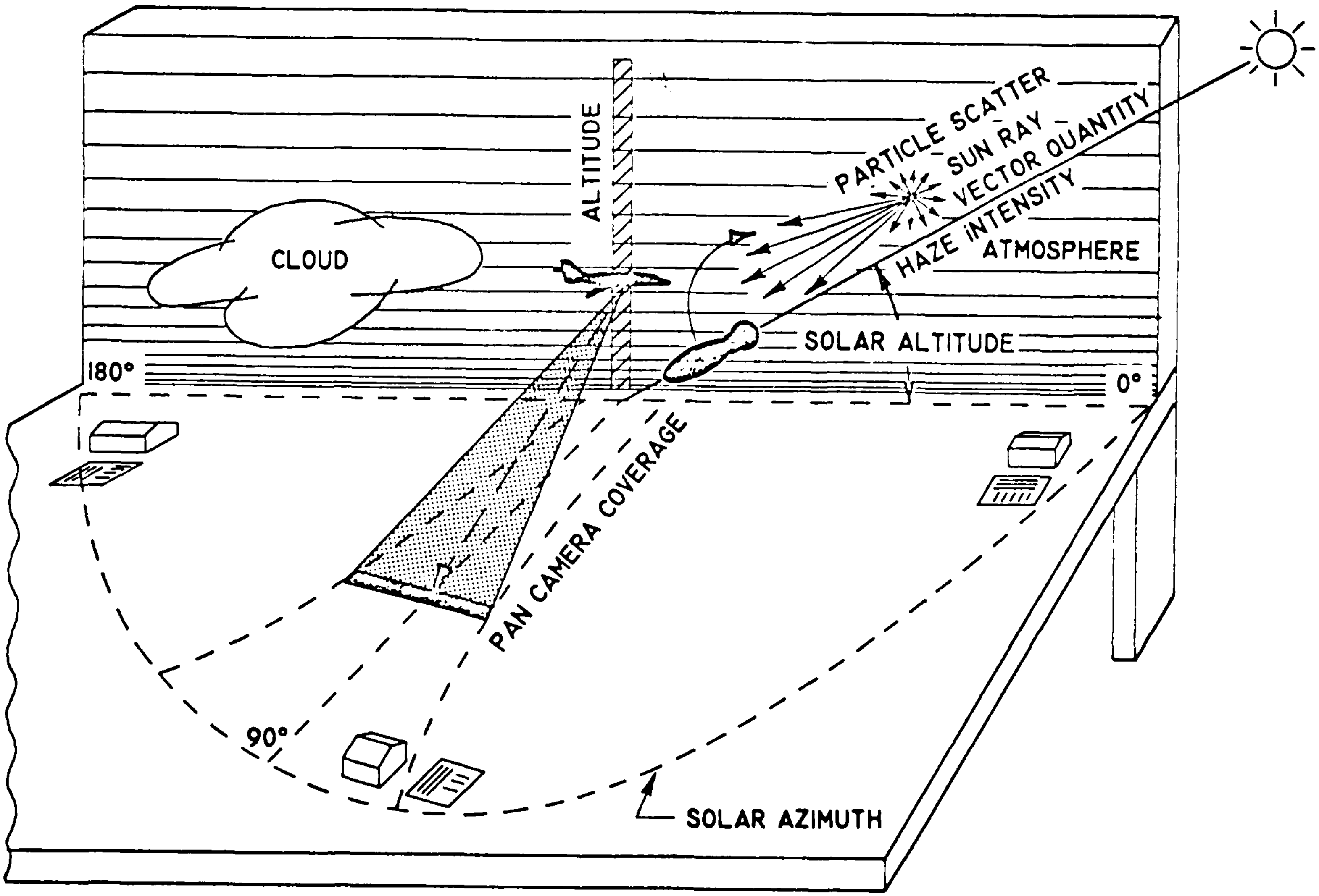


Figure A-2 Solar Illumination Factors

transmittance (τ) also varies in the same manner and is computed as a decimal. The usefulness of these terms can be shown by:

$$\overline{H}_d = \frac{\dfrac{(\overline{\sigma} N_{100}\ \tau + Na)}{100}}{4(F/\#)^2} \qquad (1)$$

Where $\overline{H}_d$ = Energy at the film ($W/cm^2/cm\lambda$)

$F/\#$ = "F" number of the lens

$\overline{\sigma}$ = Mean object reflectance (%)

N_{100} = Inherent Radiance from a perfect diffuse 100% reflectance target ($W/Sr/cm^2/cm\lambda$)

τ = Optical path transmittance (decimal)

N_a = Optical path radiance ($W/Sr/cm^2/cm\lambda$)

The above expression defines the energy available at the surface of the film ($\overline{H}_d$) as being a function of the inherent energy available ($\overline{\sigma} N_{100}$) as modified by the path transmission (τ) and haze (N_a). The individual influence of the haze, path transmittance, and image forming light can be illustrated by means of a table top model. The table top, as shown in Figure A-2, represents the earth's surface with objects in different locations with respect to the aircraft. The backboard illustrates the varying density, multilayered atmosphere and the sun is shown at a specific solar altitude above the horizon. An azimuth coordinate system has been defined with respect to the sun zenith plane. Zero degree azimuth is toward the sun, 90 degrees is at right angles, and 180 degrees is down sun. An aircraft is shown at some altitude with a sensor (panoramic camera ground coverage projection) pointing at 90 degree azimuth from the sun. One now wants to examine the change in

appearance of the objects in the scene under the varying illumination and geometric conditions. The actual appearance of the objects (with respect to brightness and contrast) depends heavily on the particle scatter. The concept of particle scatter is shown in the sun's light rays. The center dot is the particle and a lobe of back scattered light (toward sun) is shown along with a long lobe of forward scatter (away from the sun). The shape of the lobes illustrated was generated by means of plotting the computer haze energy levels in $W/Sr/cm^2/cm\lambda$ around the particle. One can more readily visualize the concept by facing the sun (30 to 40 degree solar altitude and a light hazy day) and observing the white haze in the sky around the sun and extending out to 30 to 50 degrees away from the sun. Looking toward the sun, the observer is looking into the higher intensity forward scatter. Now if the observer looks 180 degrees in back of him, he will see the sky as less white; this is due to less energy being scattered backwards. Next, looking off his shoulders at 90 degrees azimuth, he will see that the sky has the least amount of haze since he is looking into the side lobe of the particle scatter. This phenomenon can be observed in flight along with its effects upon the appearance of the ground objects. The visual spectrum imaging sensors record these variations in intensity and contrast, therefore, this phenomenon becomes very important to imagery collection and analysis.

A.2 ATMOSPHERIC HAZE MODELS

Atmospheric haze models exist that describe in mathematical terms the above effects upon sensor imagery and, therefore, are very powerful tools to

aid in mission planning and image analysis. These models compute interactions for variations (over the visual spectrum) in:

a. Solar altitudes.

b. Wavelength.

c. Haze types.

d. Aircraft altitudes.

e. Sensor look angles from vertical out to high obliques.

f. Azimuth angle from toward the sun to down sun.

g. Scene albedo (average reflectance of large area of earth's surface). In addition, specific local object and background reflectance values are overlayed on the same albedo data.

The model used by the AFAL/RWF computes the values for the inherent image forming light (N_{100}), and haze light (Na) in $W/Sr/cm^2/cm\lambda$ along with path transmission as stated above. In addition, the model also computes the ratio of Haze/(Image Light X Path Transmission). This term is called the "C" factor and relates to the object contrast transmission through the atmosphere. The number is a simple four-digit number that describes the result of the interactions of the parameters discussed above and, therefore, is very useful in determining aerial contrast ratio and contrast modulation.

A.3 APPLICATION OF 'C' FACTOR

The practical analysis of target data has been a longstanding problem which now has a workable solution using the various atmospheric haze models. For example, the model used by AFAL/RWF includes the haze (Na), inherent image forming light (N_{100}), path transmittance (τ) and the ratio ($Na/N_{100}\tau$) x 100 which is the "C" factor. These terms are used to compute the aerial contrast ratio, where aerial contrast ratio is defined as:

$$C_r = \frac{R_b + \frac{N_a}{N_{100}\tau} \times 100}{R_t + \frac{N_a}{N_{100}\tau} \times 100} \qquad (2)$$

or

$$C_r = \frac{R_b + C}{R_t + C} \qquad (3)$$

Where

R_t = Reflectance object (percent)

R_b = Reflectance background (percent)

C = $(Na/N_{100}\tau) \times 100$

These equations combine the power of the modern digital computer (which computes values of Na, N_{100}, τ, and C for any given situation) with the simplicity of a small equation readily solved by an analyst using a pocket calculator. The above equations can be rearranged to solve for the "C" factor for specific values of R_t, R_b, and C_r.

$$C = \frac{R_b - C_r R_t}{C_r - 1} \tag{4}$$

Equations (2), (3), and (4) are useful for examining a range of C, C_r, R_t, and R_b values that can be compiled into graphic form as illustrated in Figure A-3. This figure shows the relationship between "C" factor and one value of R_t (olive drab paint = 11% reflectance) and four values of R_b (sand = 35%, dry vegetation = 26%, asphalt = 19%, and green vegetation = 15% reflectance). Using equation (3) and the R_t of OD paint = 11%, R_b of sand = 35% and "C" = 1.0 provides an aerial contrast ratio of 3:0. When the value of "C" is increased to 10 and then 100, the aerial contrast ratios become approximately 2.18:1 and 1.22:1, respectively. The other curves follow in the same manner. An analyst can also work the problem in reverse and ask the question, "What would be the "C" factor for an aerial contrast ratio of 2.0:1 for the OD paint and sand case?" This is solved by intersecting the curve (11% and 35%) with a line up from the aerial contrast ratio of 2.0:1 and then reading over the "C" factor scale where the value is 14.0. This number would then be related to haze model data or atmospheric data provided by test data. Observing the trends of the curves of Figure A-3, one concludes that a "C" factor of 1.0 very nearly represents ground contrast and that contrast is reduced as the "C" factor increases in value. For example, when the values of "C" become larger than 60, contrast loss becomes very severe.

The curves shown in Figure A-3 illustrate the usefulness of the "C" factor concept for assessing contrast variations and, in addition, the usefulness of this type of graphic presentation for both flight planning and post flight analysis. Where many different values of object and background reflectance values must be examined in conjunction with a specific aerial

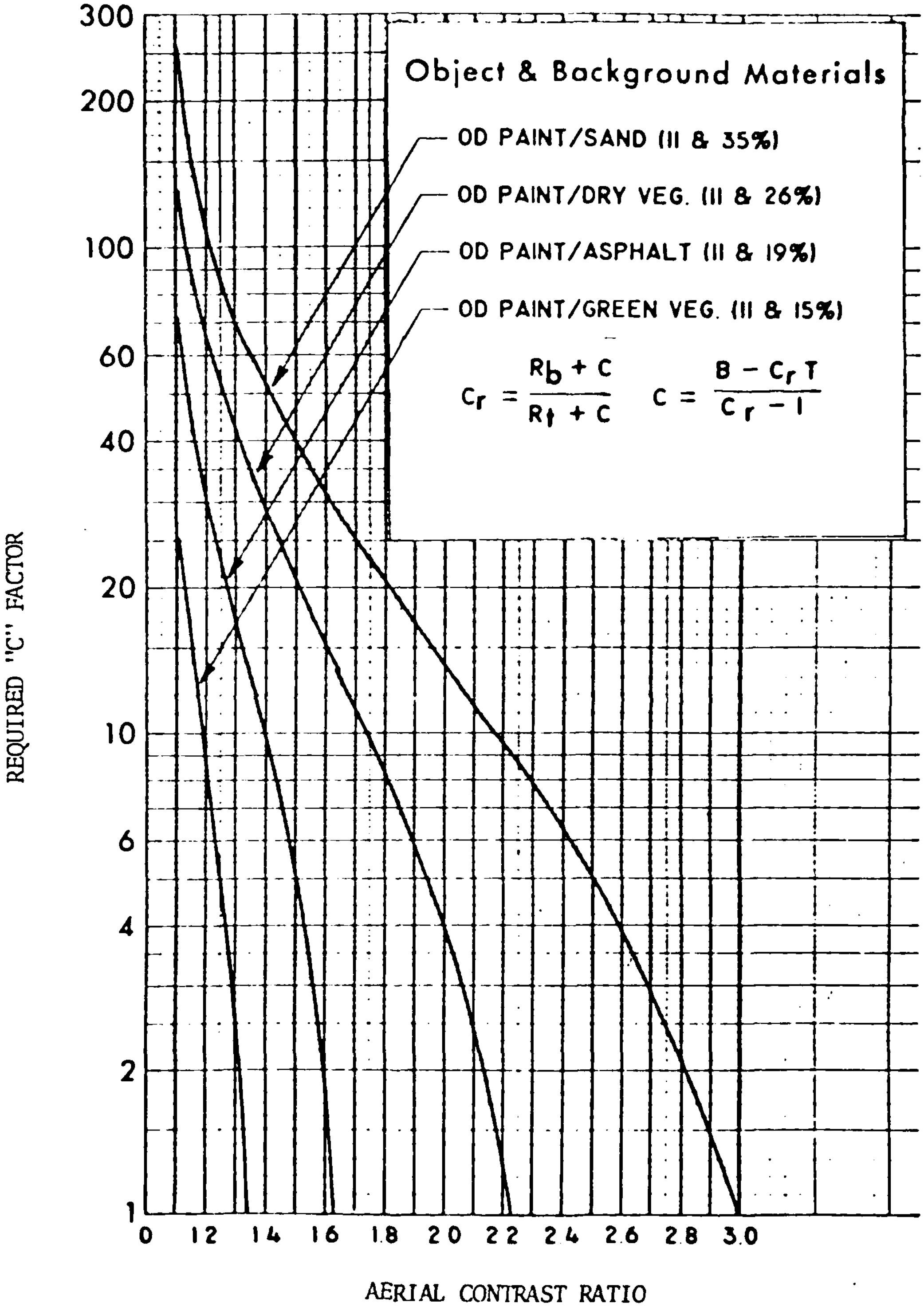


FIGURE A-3. "C" FACTOR VS AERIAL CONTRAST RATIO

contrast ratio, the data can be assembled in the manner illustrated in Figures A-4 and A-5. These curves were compiled by means of equation (4) for aerial contrast ratio values of 3.0:1 and 2.0:1, for the object and background reflectance. The vertical lines are the object reflectance and the curved lines are the background reflectance. The maximum allowable "C" factor is read opposite the intersection of the object and background reflectance curves. For example, using Figure A-4 with object (10%) and background (40%) the maximum allowable value of "C" is 5.0 for retaining a 3:1 aerial contrast ratio. Applying the same reflectance values in Figure A-6 (2:1 aerial contrast) one finds that the allowable value of "C" has increased to 20.

The data presented in Figures A-4, A-5, and equation (3) indicate that for a given value of "C" there are many different combinations of target (R_t) and background (R_b) reflectance values that will provide the same contrast ratio. This can be illustrated by Figures A-6 and A-7 where equation (3) was used to compute contrast ratio as a function of C for ground contrast ratios of 2:1 and 4.5:1. This was accomplished for a series of target and background reflectance values ranging from very dark target and background combinations to very bright target and backgrounds. Observing Figure A-6, the left side of the family of curves are very low reflectance (5/2.5 percent) while the right side is very high reflectance (90/45 percent). The low reflectance values suffer contrast loss at much lower values of "C" while the brighter targets retain their contrast under considerably larger values of "C". The average reflectance value is included for use in estimating the brightness of the target and background.

Practical use can be made of the curves presented above to determine the aerial contrast value for given target and background reflectance values,

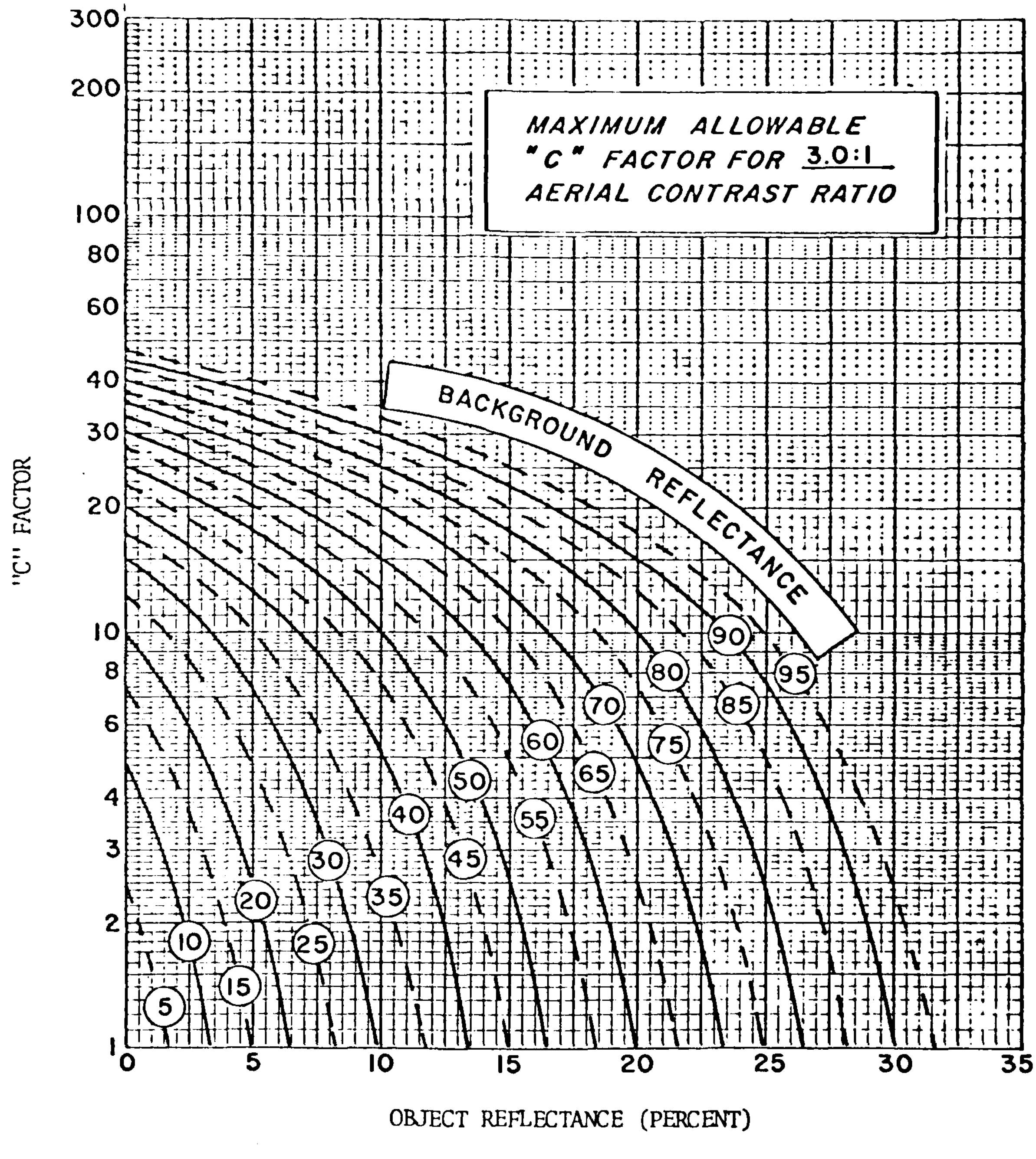


FIGURE A-4. "C" FACTOR SUMMARY FOR 3:1 CONTRAST RATIO

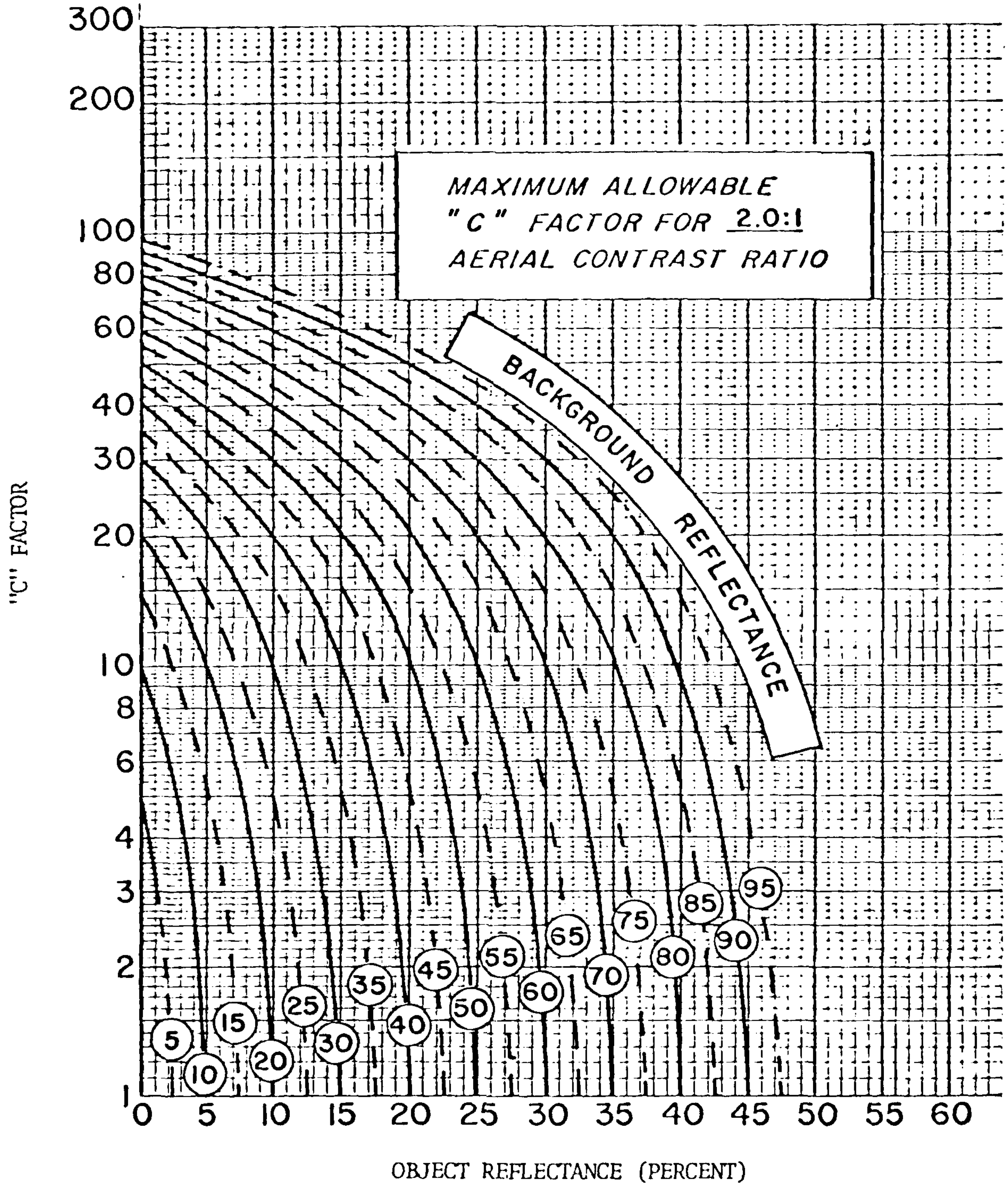


FIGURE A-5. "C" FACTOR SUMMARY FOR 2:1 AERIAL CONTRAST RATIO

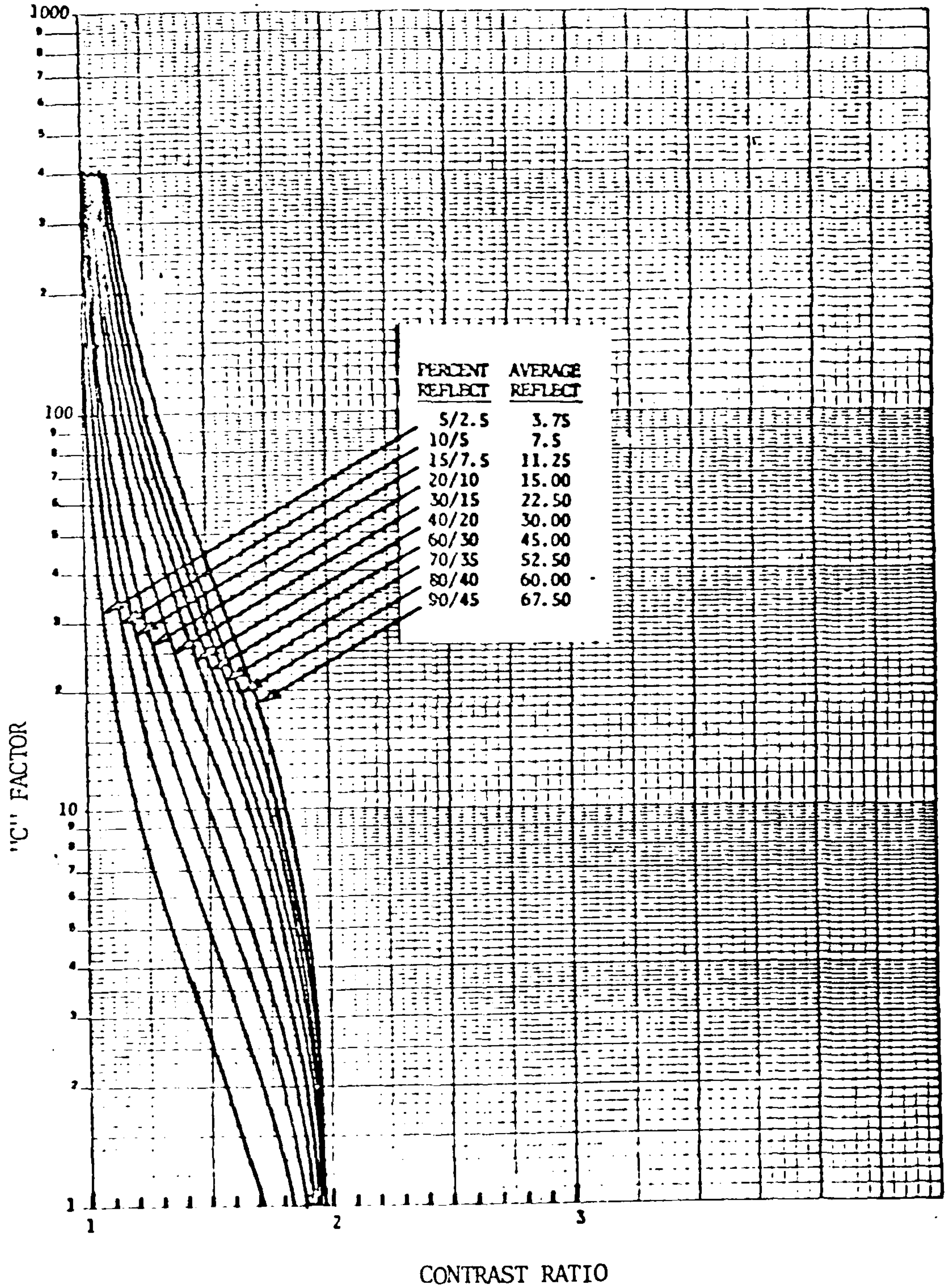


Figure A-6 "C" Factor Versus Aerial Contrast for 2:1 Ground Contrast Ratio

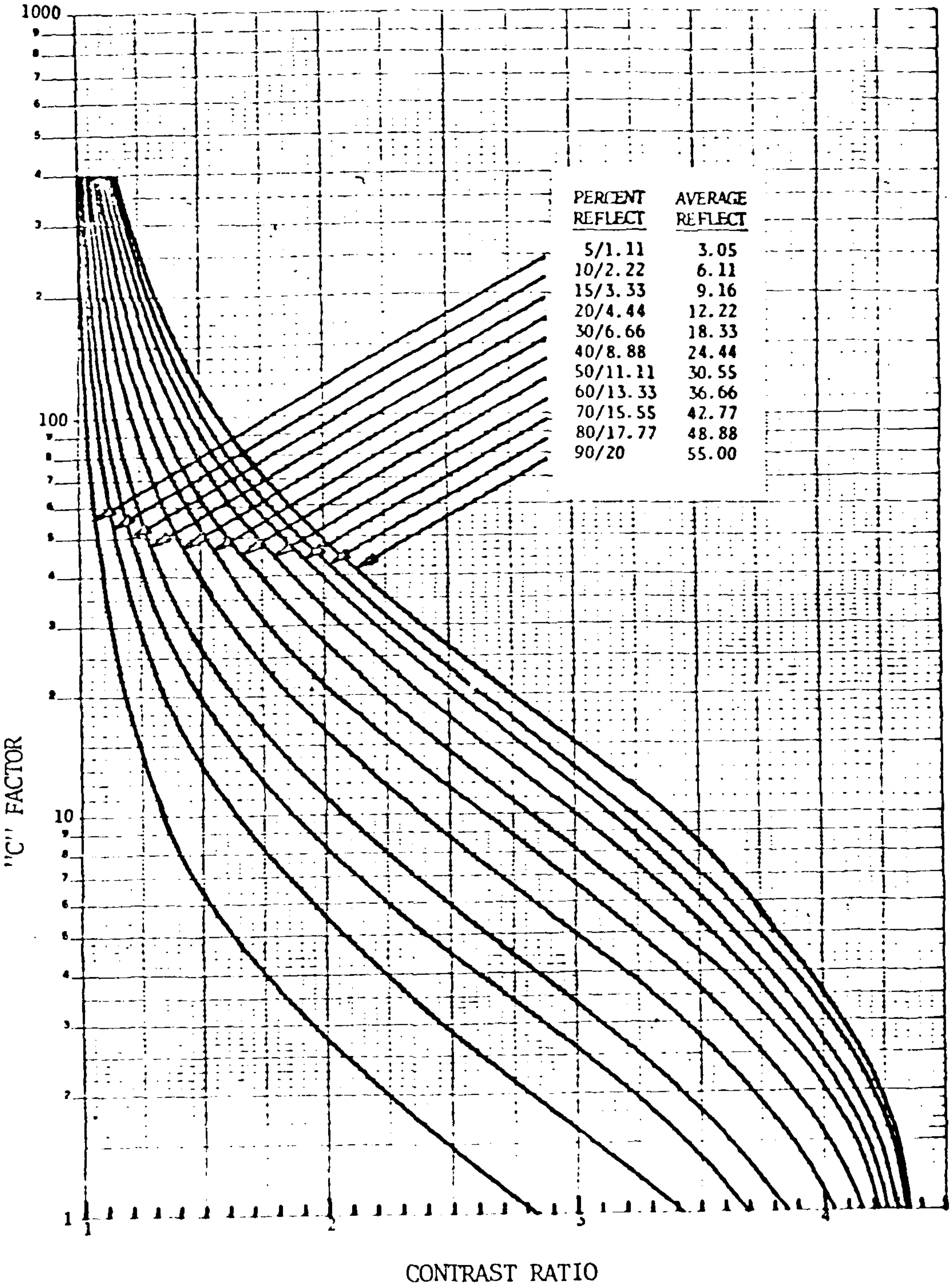


Figure A-7 "C" Factor Versus Aerial Contrast for 4.5:1 Ground Contrast Ratio

where "C" factor values are either computed or derived from flight data. In addition, if and only if reliable values of ground reflectance are available and aerial contrast can be measured, then from these known data the "C" factor can be obtained for the atmospheric path.

In summary, the target background reflectance/contrast curves illustrate the ability of the energy reflected from the target and background to overcome the degradation effect of haze. In addition, they illustrate the usefulness of the "C" factor term in problem solving as a means to bridge between the computer models, measured flight, and laboratory data.

APPENDIX B

CONTRAST RESTORATION SOFTWARE

B.1 INTRODUCTION

This appendix contains descriptions of the software developed for use with the Automatic Feature Extraction System at Rome Air Development Center. This software consists of two FORTRAN programs: AREFL and AREC. When properly used, these programs restore the natural contrast to aerial imagery degraded by atmospheric haze. To do this, these programs first develop a calibrated relationship of film density to ground reflectance for a specified image. This relationship is then used to convert an image recorded on magnetic tape in density units to a restored image recorded on magnetic tape in units of reflectance. This restored image, when displayed, has the appearance of the scene with little atmospheric degradation.

B.2 PROGRAM AREFL

The program AREFL is the primary program of the contrast restoration process (See Figure B-1). This program develops the calibrated relationship between film density and ground reflectance for a specified image. The calibration is based on the approximate reflectance values of certain identifiable objects in the image such as roads, runways, grass, sand, dirt, and paint.

AREFL begins by reading a density and reflectance value for each calibration object in the image and the D Log E curve for the film. These are used to compute the linear regression coefficients that relate relative exposure to ground reflectance. The program then uses these coefficients and

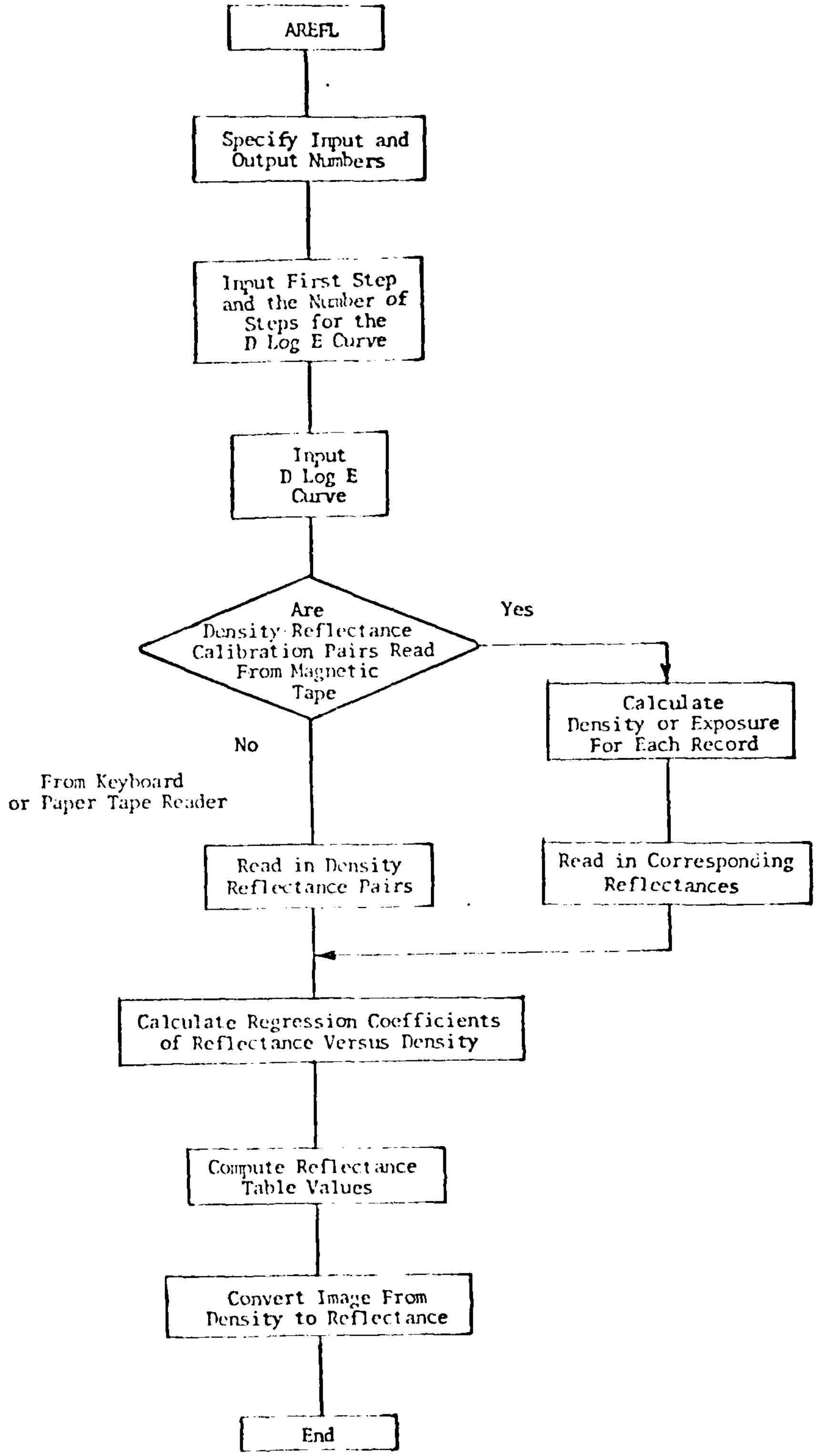


FIGURE B-1. PROGRAM AREFL FLOW DIAGRAM

the D Log E curve to construct a table of reflectance values corresponding to all the possible density values of the image. This table is then used to convert the density image into a reflectance image.

B.2.1 PROGRAM-AREC

AREC develops the calibration density-averaged values for use in program AREF. These density values are average values for a number of specified calibration areas whose size and shape are defined by a set of cartesian coordinates.

AREC begins by reading in the D Log E curve of the film and a number of coordinates to describe each calibration area as noted in Figure B-2. After finding a calibration area on the tape of the image, each point that is in the specified area is averaged as a density value or converted to exposure and averaged as an exposure value.

In order to determine if a pixel is within an area, an imaginary rectangle is developed which encloses the area as illustrated in Figure B-3. This rectangle's coordinates consist of the maximum and minimum X and Y values used to specify the calibration area. Along one axis of the rectangle is the scan direction and along the other is the axis perpendicular to the scan. Any scan outside the rectangle is rejected. For the remaining scans that pass through the rectangle, those points at the beginning and end of each scan that are outside the rectangle are also rejected.

The remaining points are tested to determine which are within the boundaries of the specified area of the cartesian coordinates by the subroutine NTIN (Reference Figures B-3 and B-4). This routine constructs an

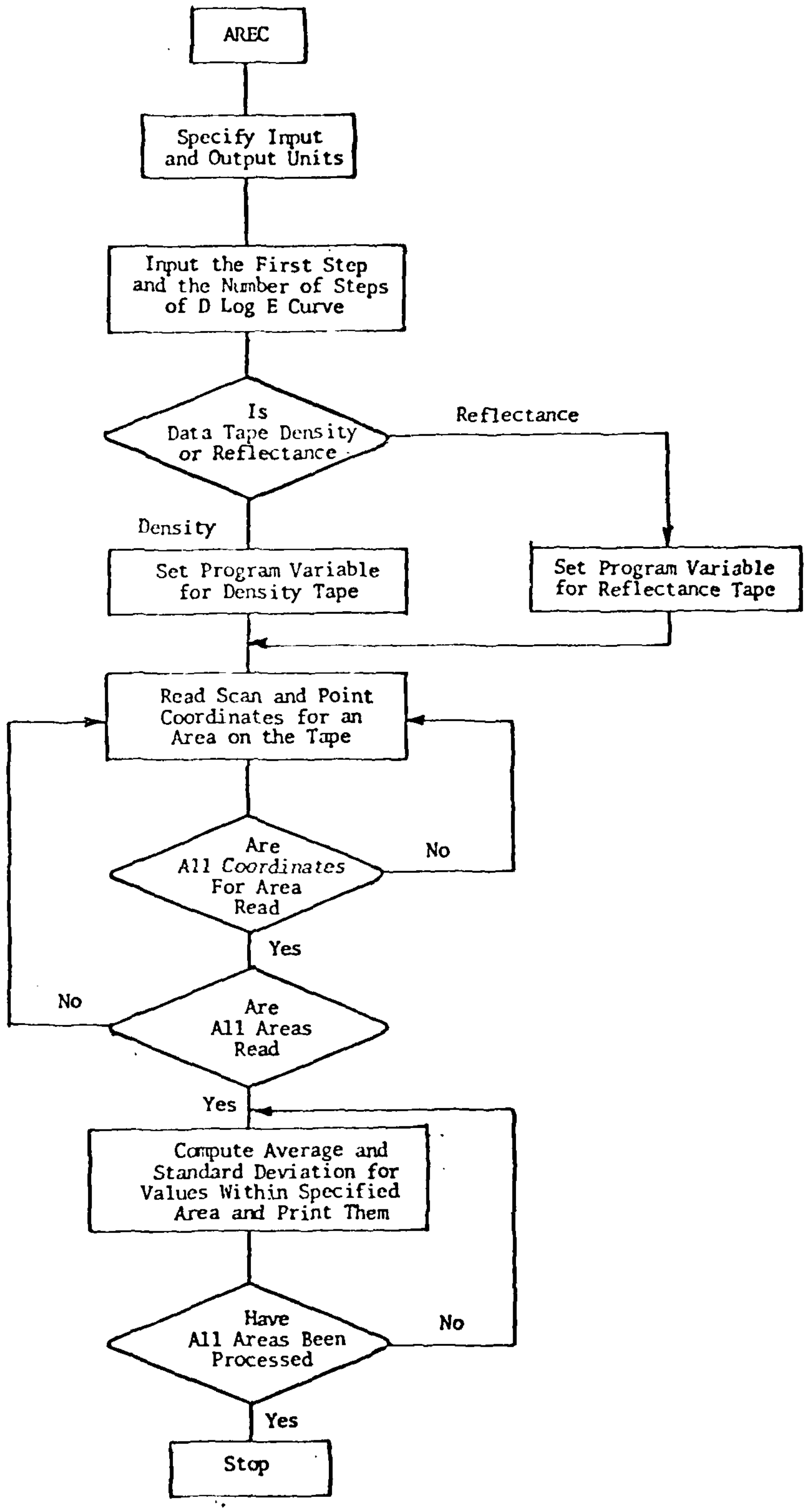
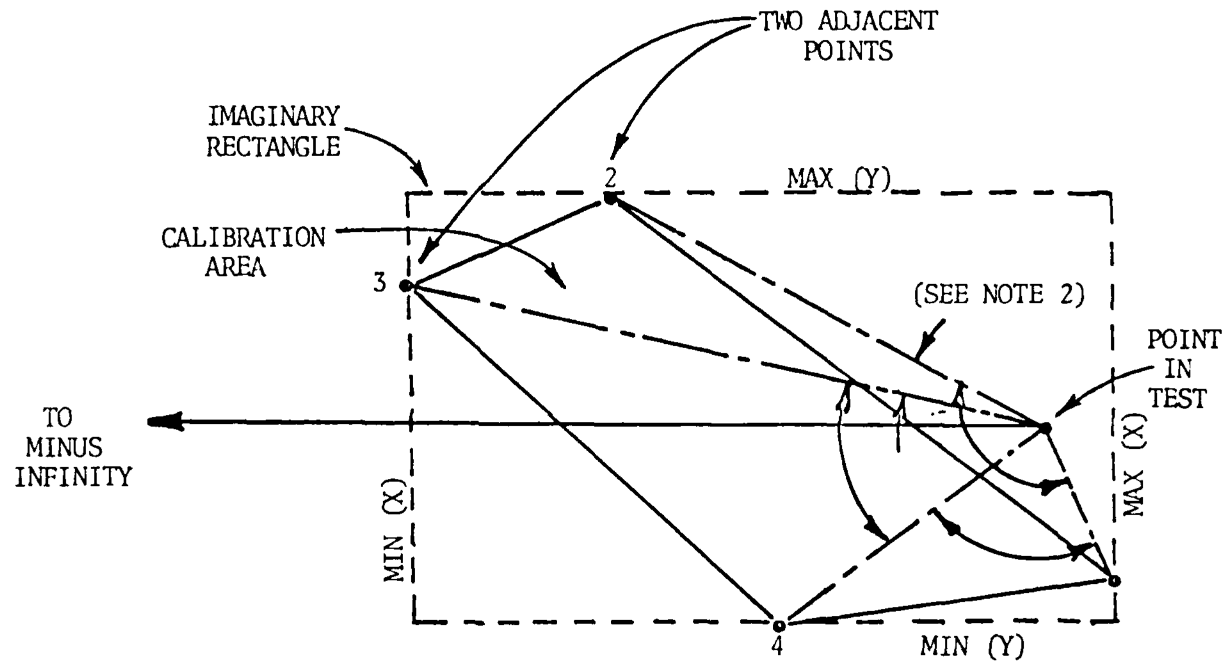


FIGURE B-2. PROGRAM AREC FLOW DIAGRAM



NOTE 1. POINTS 1, 2, 3 AND 4 ARE THE CARTESIAN COORDINATES OF THE CALIBRATION AREA

2. ALL ANGLES ARE LESS THAN 180°

FIGURE B-3. NTIN SUBROUTINE TEST OF A POINT POSITION

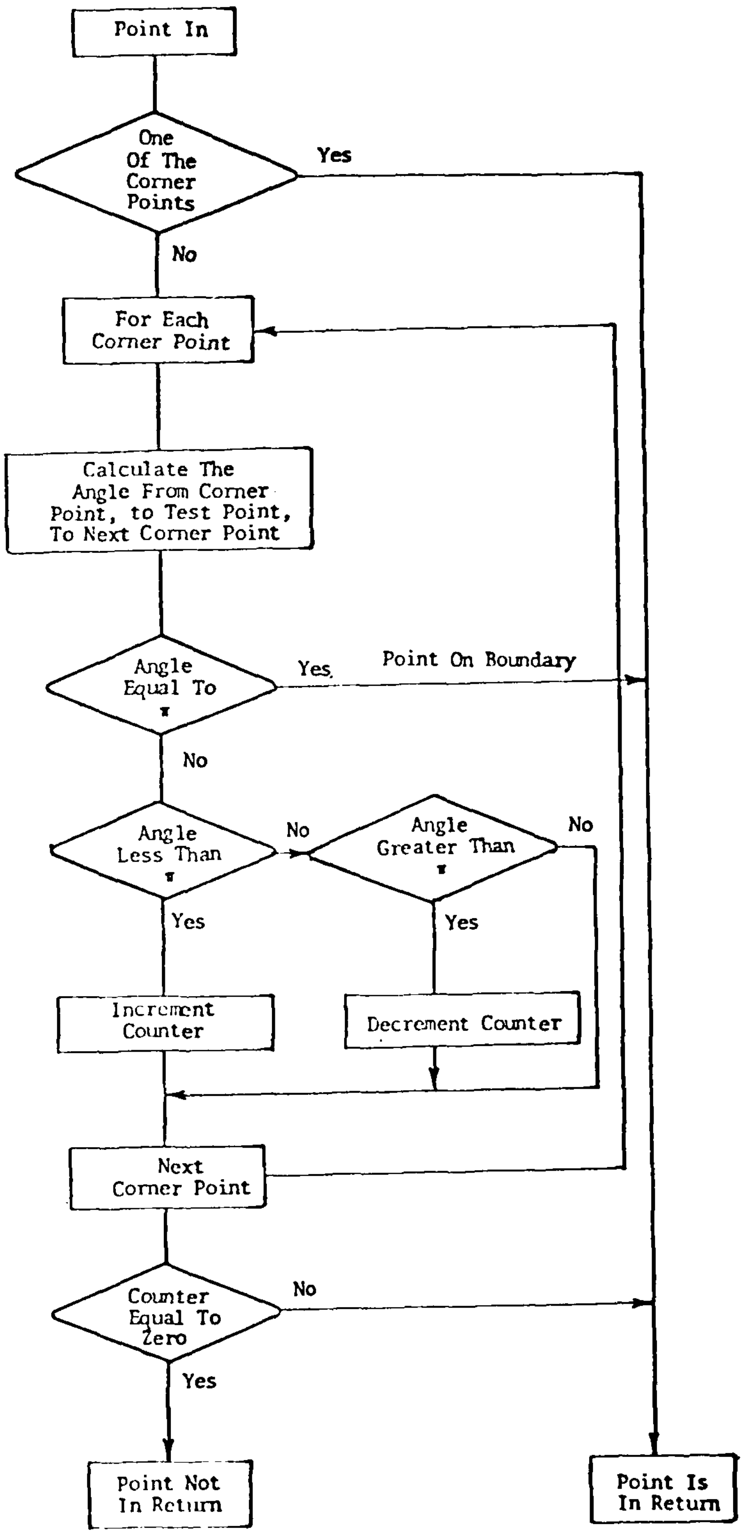


FIGURE B-4. SUBROUTINE NTIN FLOW DIAGRAM

imaginary line from the point being tested to minus infinity, and an angle that is formed by constructing two lines from two adjacent coordinates of the specified boundary to the point being tested. The apex of this angle is then at the point being tested (See Figure B-3). A count is kept of the number of times this line is within this acute angle over all possible angles described. If the final count is even (including zero), the point is outside the boundary; if the count is odd, it is within the boundary.

APPENDIX C

REFLECTANCE DATA SUMMARY

C.1 TARGET SIGNATURE TABLES

Target signatures (as they apply to aerial photography) are the spectral reflectance curves of a target in the visible and near- infrared wavelength regions, that is, those regions to which film is sensitive. They can be useful for many analysis problems such as selecting filters to enable one to filter out a background or emphasize a target's dominant reflectance, thereby improving contrast and, hence, system resolution. In this report they are used to provide the reflectance of known objects as a portion of the contrast restoration technique.

The signatures which appear as tables and curves in this appendix are divided into several broad classes and include paints, metals, soil, grass, and other vegetation. Computed reflectance data for selected manmade materials are given for three wavelengths (0.5, 0.62, and 1.0) and three wavelength bands (0.4-1.0, 0.52-0.72, and 0.58-0.72) in Table C-1. The three wavelength bands represent the spectral regions covered with no filter, a Wratten 12 filter, and a Wratten 25 filter, respectively. The data in Table C-2 are composite reflectance values for use with the spectral band of photography. The data includes the mean and plus or minus one standard deviation values for the natural material listed. The individual reflectance values for the selected materials, over the wavelength band from ..4 to 1.2 microns, are tabulated in Table C-3. These data provide a means to compute the reflectance values for other wavelength bands. It should be noted that in the application of the target reflectance data the general background albedo

MATERIAL	WAVELENGTH (MICRONS) .4-1.0	.52-.72	.58-.72	.5	.62	1.0
White paint on metal, Military Standard 37875	74.5	79.5	78.0	83.5	79.0	----
White paint on aluminum Substrate, 3M Velvet	84.5	88.0	88.0	87.0	88.0	----
Commercially pure aluminum treated 300 Hrs. at 600 F.	69.0	70.5	70.5	69.0	70.5	72.0
Weathered aluminum 20,000 Hrs. on DC-6	39.5	37.0	37.0	37.0	37.0	55.0
Galvanized iron commercially finished	37.0	41.0	40.5	44.0	41.5	27.0
Rusty Iron	8.0	9.0	9.5	7.0	9.5	----
Roofing felt paper tar paper Black	4.0	4.0	4.0	4.0	4.0	4.0
Roofing asphalt aged	17.5	19.0	19.0	17.0	19.0	----
Aged cement	37.0	----	----	----	----	37.0
Concrete aged/wet/dry	39.5	38.0	39.0	35.0	38.0	46.0
Concrete Willow Run Airport 24 Years	22.5	23.0	24.0	17.0	24.0	25.5
Tar road bed, black	8.0	9.0	10.0	7.0	10.0	----

---- = No Data

Table C-1. REFLECTANCE VALUES (PERCENT) FOR SELECTED MAN-MADE MATERIALS

TABLE C-2

COMPOSITE TARGET SIGNATURE REFLECTANCE

DATA FOR B&W FILM AND #12 OR 25 FILTER

MATERIAL	FILTER NUMBER	PERCENT REFLECTANCE +1 Sigma	Average	-1 Sigma
All Green Vegetation	12	21	15	10
	25	21	15	10
Brown Dead Dry Vegetation	12	30	23	17
	25	33	26	19
Tree Bark	12	21	13	8
	25	22	14	9
Dry Sandy Soils	12	40	34	28
	25	43	37	32
Wet Sandy Soils	12	23	18	13
	25	25	20	15
Dry Loam Soils	12	20	14	10
	25	22	16	11
Wet Loam Soils	12	11	8	4
	25	12	9	5
All Dry Soils	12	39	24	10
	25	42	26	11
All Wet Soils	12	22	13	4
	25	24	14	4
All Wet & Dry Soils	12	32	20	6
	25	35	22	7

Wave Microns	Green Veg.	Dried Veg.	Dry Sandy Soil	Wet Sandy Soil	Dry Loam	Wet Loam	All Dry Soil	All Wet Soil	New Cement	Weath. Cement	All Soils	Tree Bark
.4	5	6	13	7	5	3	9	4	24	9	7	9
.45	7.5	8	16	8	6	4	11	5	30	14	8	10
.5	8.5	12	19	9	7	4	14	6	35	17	10	10
.55	16	16	26	12	10	6	18	9	37	21	15	12
.6	13	21	36	18	15	8	25	14	38	23	19	14
.65	10	26	39	21	18	9	28	16	39	24	22	16
.7	18	24	42	23	20	11	31	17	41	25	24	20
.75	50	38	45	25	23	12	34	20	43	26	27	23
.8	54	42	47	26	23	13	36	20	45	27	29	26
.85	54	46	47	27	23	12	36	20	45	27	28	29
.9	55	49	48	27	23	13	37	20	43	26	29	30
.95	54	52	50	28	24	13	38	21	43	26	30	32
1.0	54	54	52	29	25	15	39	22	45	26	31	34
1.05	55	55	54	31	26	15	40	24	45	26	32	36
1.1	54	60	55	32	28	16	41	24	46	27	33	36
1.15	50	64	54	31	29	17	42	25	46	27	34	36
1.2	49	65	54	31	29	18	42	25	47	27	34	37
Ave.	35.7	38.1	41	22.6	19.6	11.1	30.6	17.1	40.7	23.4	24.2	24.1

Frenchmans Flat = 25.4
Desert = 37.6

TABLE C-3

MEAN REFLECTANCE OF TERRAIN MATERIALS (PERCENT)

also affects the aerial contrast ratio. Higher background reflectances will increase the path-radiance contribution to exposure, thereby reducing contrast. Table C-4 gives several common background albedos.

C.2 TARGET SIGNATURE CURVES

The reflectance curve data for the selected natural materials are shown in Figures C-1 through C-9. These curves show the significant changes in reflectance as a function of wavelength and the magnitude of difference in reflectance of the material samples measured.

The composite green vegetation curve in Figure C-1 is composed of approximately 1000 individual curves*. It should be noted that these curves are made up of deciduous, palmaceous, and coniferous tree leaves or needles, and also crops and grasses. Therefore, it cannot be assumed that this composite is entirely representative of all green vegetation, although it is a major improvement from the past data values. There are two basic reasons why the assumption may not be valid. First, all of the vegetation curves are from vegetation in the United States. Therefore, it is possible that it does not represent worldwide data. Second, the number of types of vegetation is only a small percentage of United States species, and a smaller portion of worldwide types; the average is not weighted to reflect the material's relative frequency of occurrence. Even though these qualifications of the data have been made, they are only mentioned as a precaution to the user of the composite curve, and are not intended to make the data appear invalid or discourage use of the data. Most of these individual curves are similar in shape which indicates that all green vegetation would probably follow these

* The central curve represents the mean of all samples, while the upper and lower curves are plus and minus one standard deviation values.

TABLE C-4

COMMON BACKGROUND ALBEDO

General	0.15
Water	0.04-0.05
Sand	0.40
Fresh Snow	0.60
Forest	0.10
Scattered Clouds (2/8-4/8)	0.35-0.40
Broken Clouds (4/8-6/8)	0.40-0.60

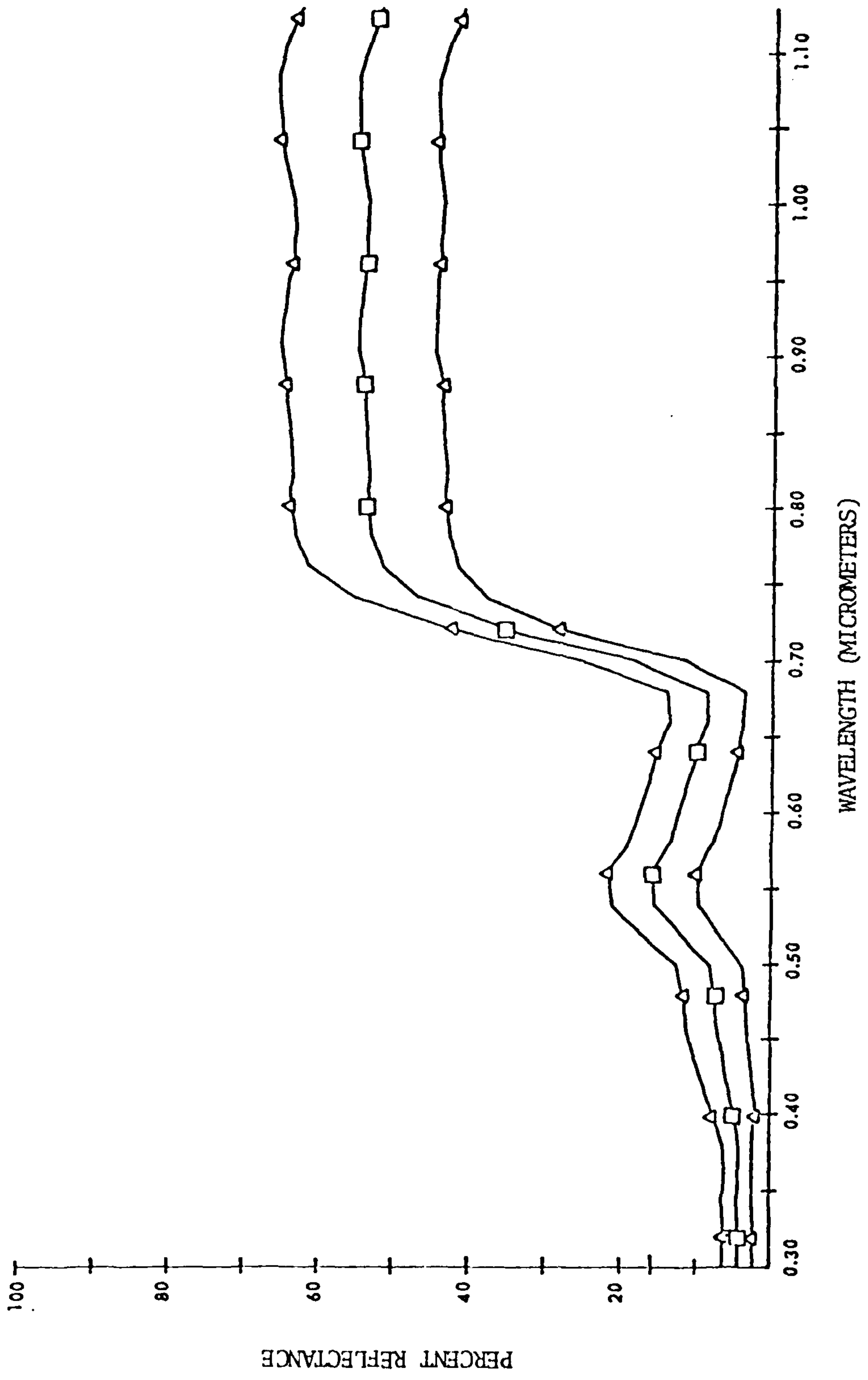


FIGURE C-1. GREEN VEGETATION

same general curve shapes. This is also true for the following curves where limited samples exist.

The composite brown, dead, dried vegetation curve in Figure C-2 is composed of 19 individual curves. These curves are made up of only deciduous tree leaves found in the United States, therefore, these few curves may not be valid for all brown, dead, dried vegetation. It is neither based on worldwide data nor is it composed of other types of dried vegetation such as crops, grasses, weeds, or other seasonal types of undergrowth.

There are seven composite soil curves furnished. The first in Figure C-3 is dry sandy soil composed of nine curves. The second in Figure C-4 is wet sandy soil composed of nine curves. The third in Figure C-5 is dry loam soil composed of sixteen curves. The fourth in Figure C-6 is wet loam soil composed of fourteen curves. The fifth curve in Figure C-7 is composed of all the dry soils in the data bank (96), while the sixth curve in Figure C-8 is composed of all wet soils (90). The last curve in this sequence, Figure C-9, is a composite of all soil curves in the data bank and is compiled from 207 curves. Although these data are primarily from continental United States, they also include soils from Hawaii, Puerto Rico, Cuba, and Honduras. The first four sets were shown to reflect relatively "tight" sets and were based upon granular size of the soils, that is, sand or loam, and not by chemical composition. At this time it is not known what amplitudes of the changes in soil can be attributed to either granule size or chemical composition. A further comparison of the first soil curve with the second, and the third soil curve with the fourth, will show that the moisture content influences the reflectance of both of these soils. Documentation on these curves did not

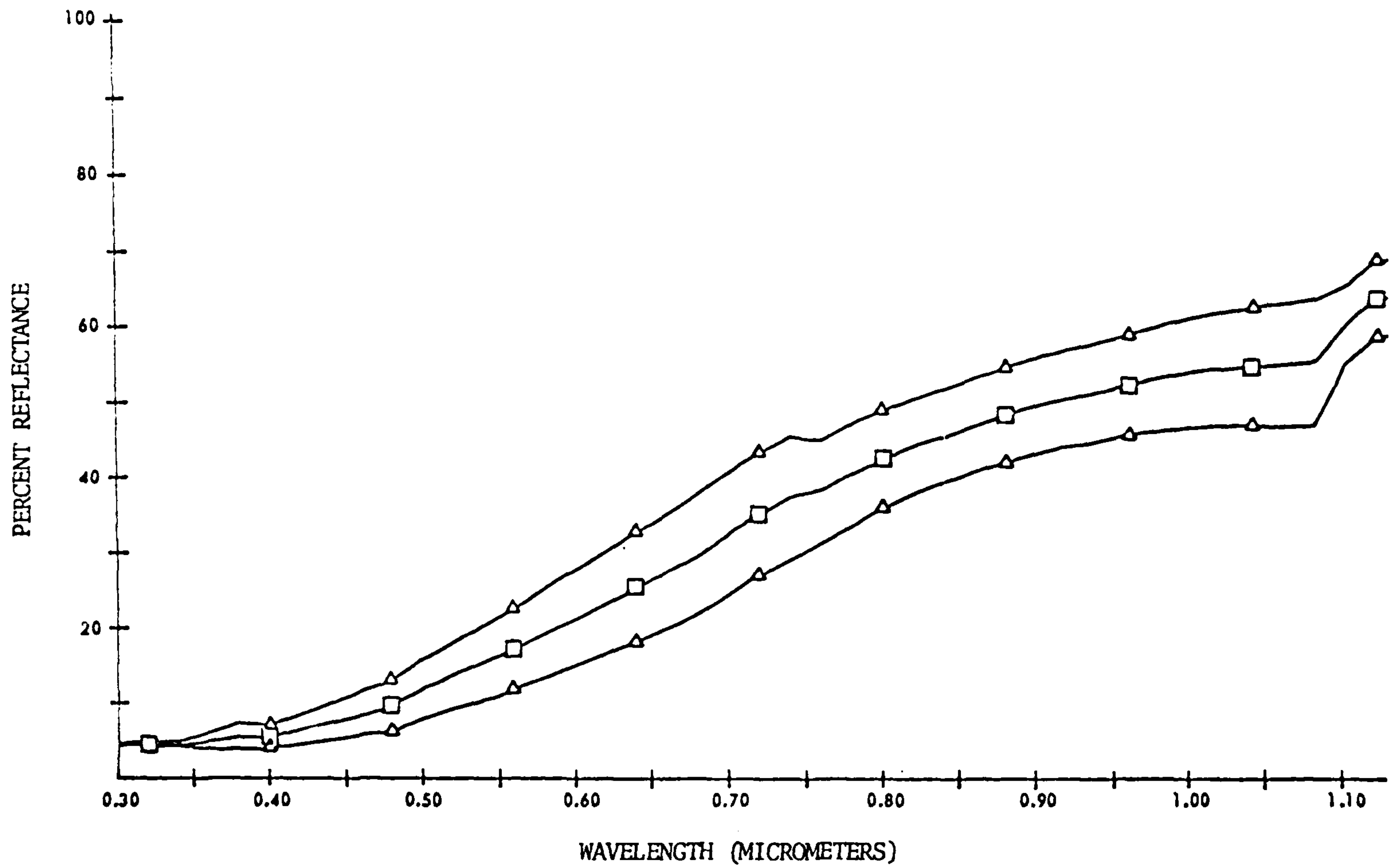


FIGURE C-2. DRIED VEGETATION

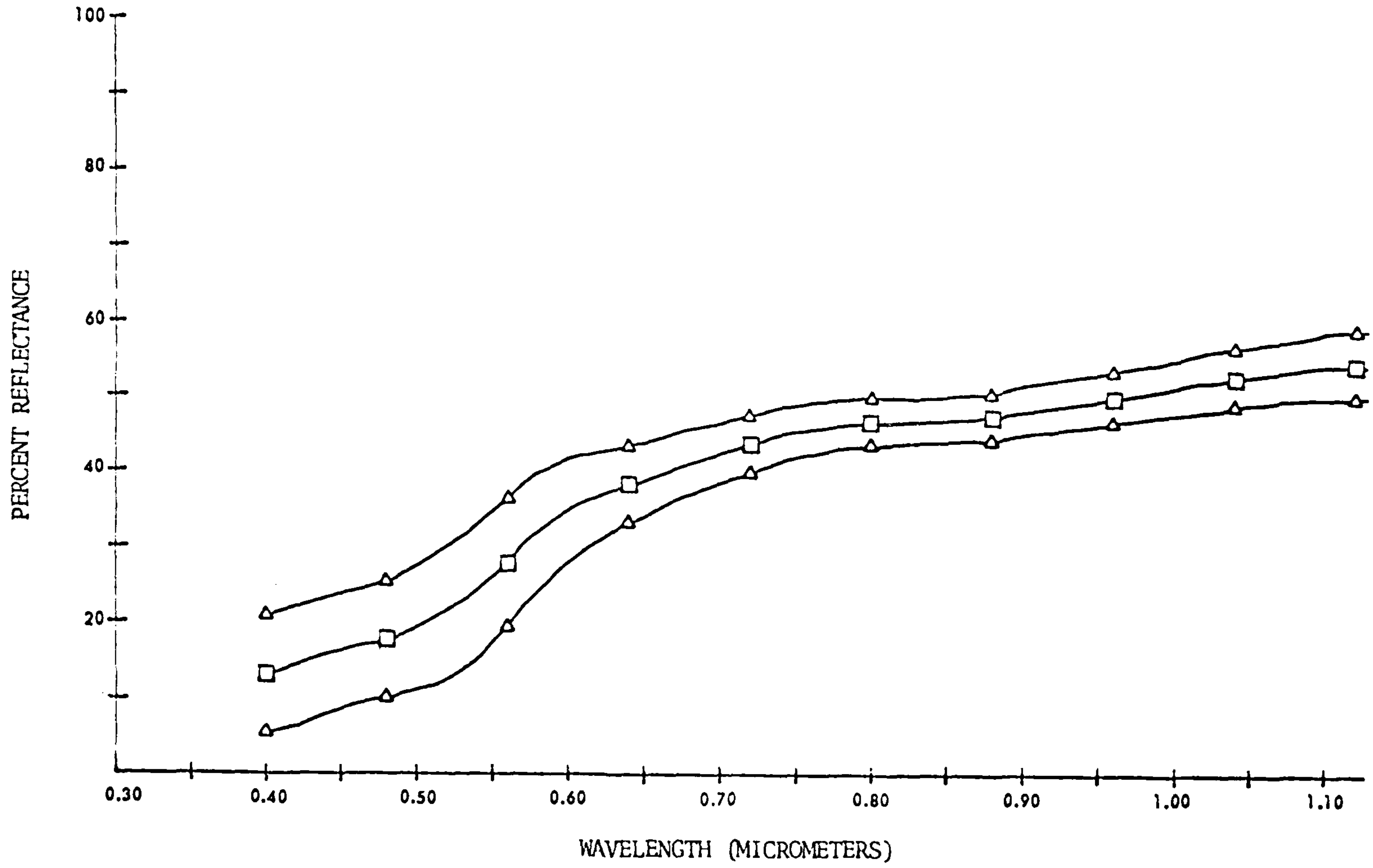


FIGURE C-3. DRY SANDY SOIL

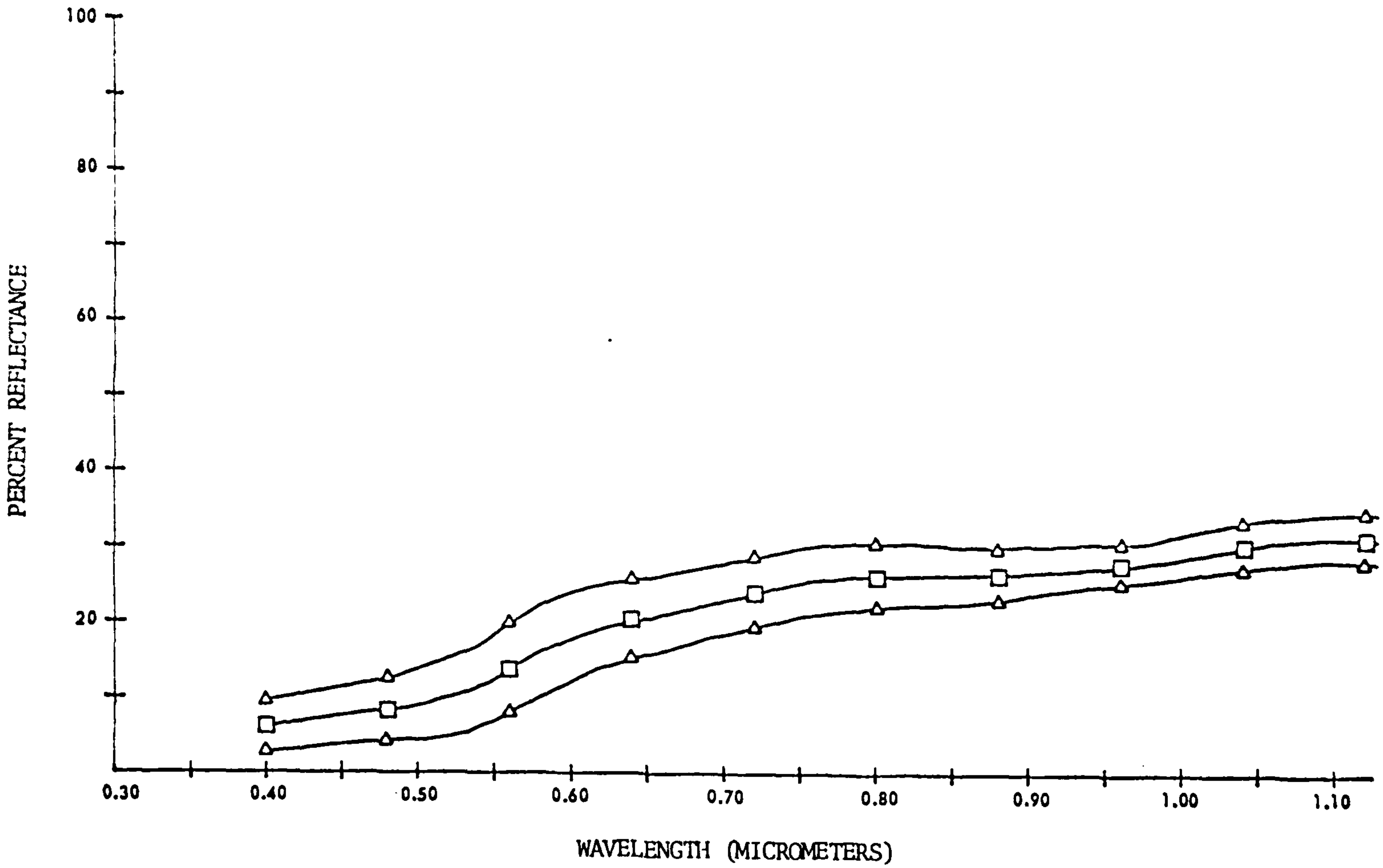


FIGURE C-4. WET SANDY SOIL

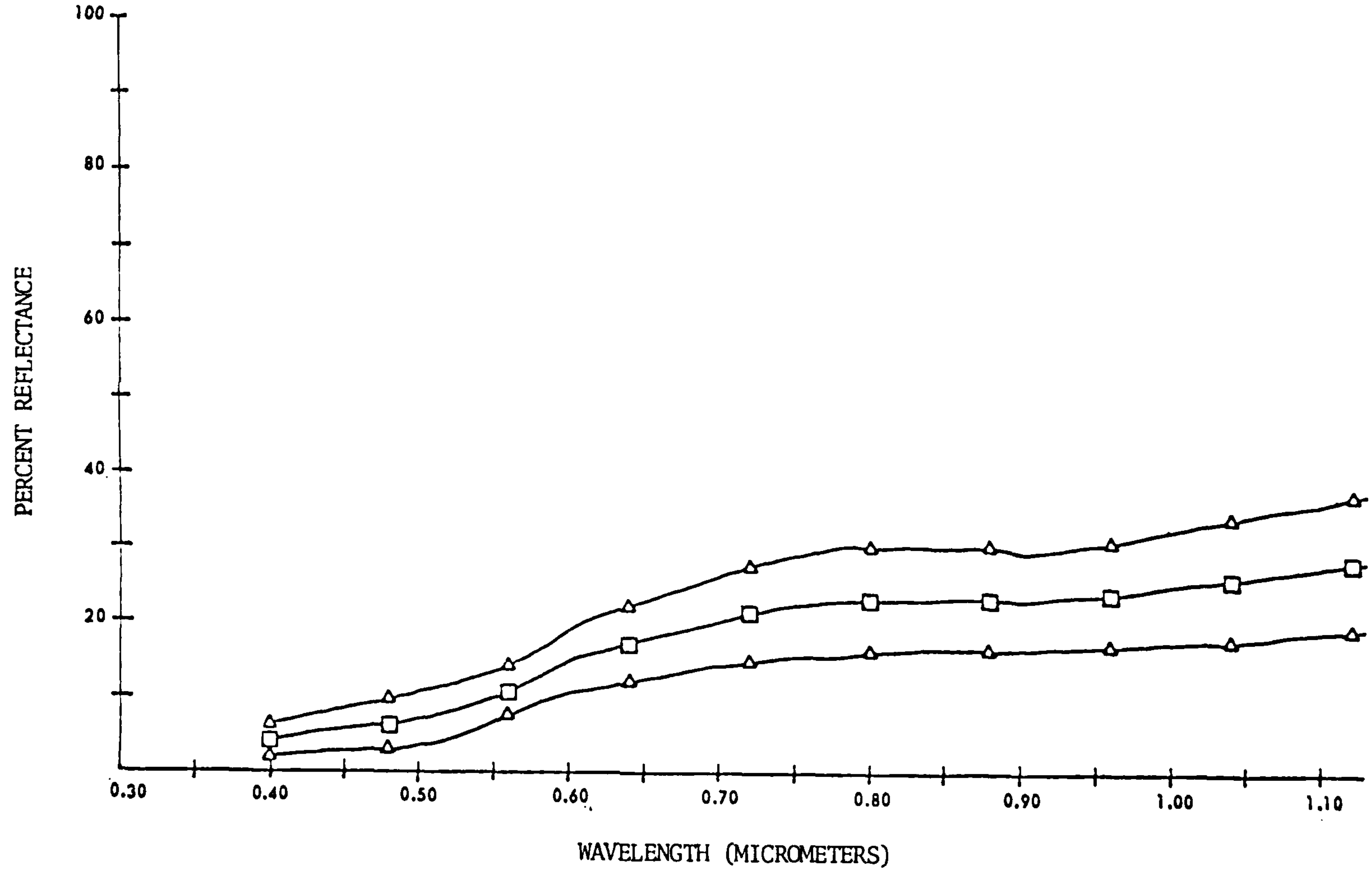


FIGURE C-5. DRY LOAM

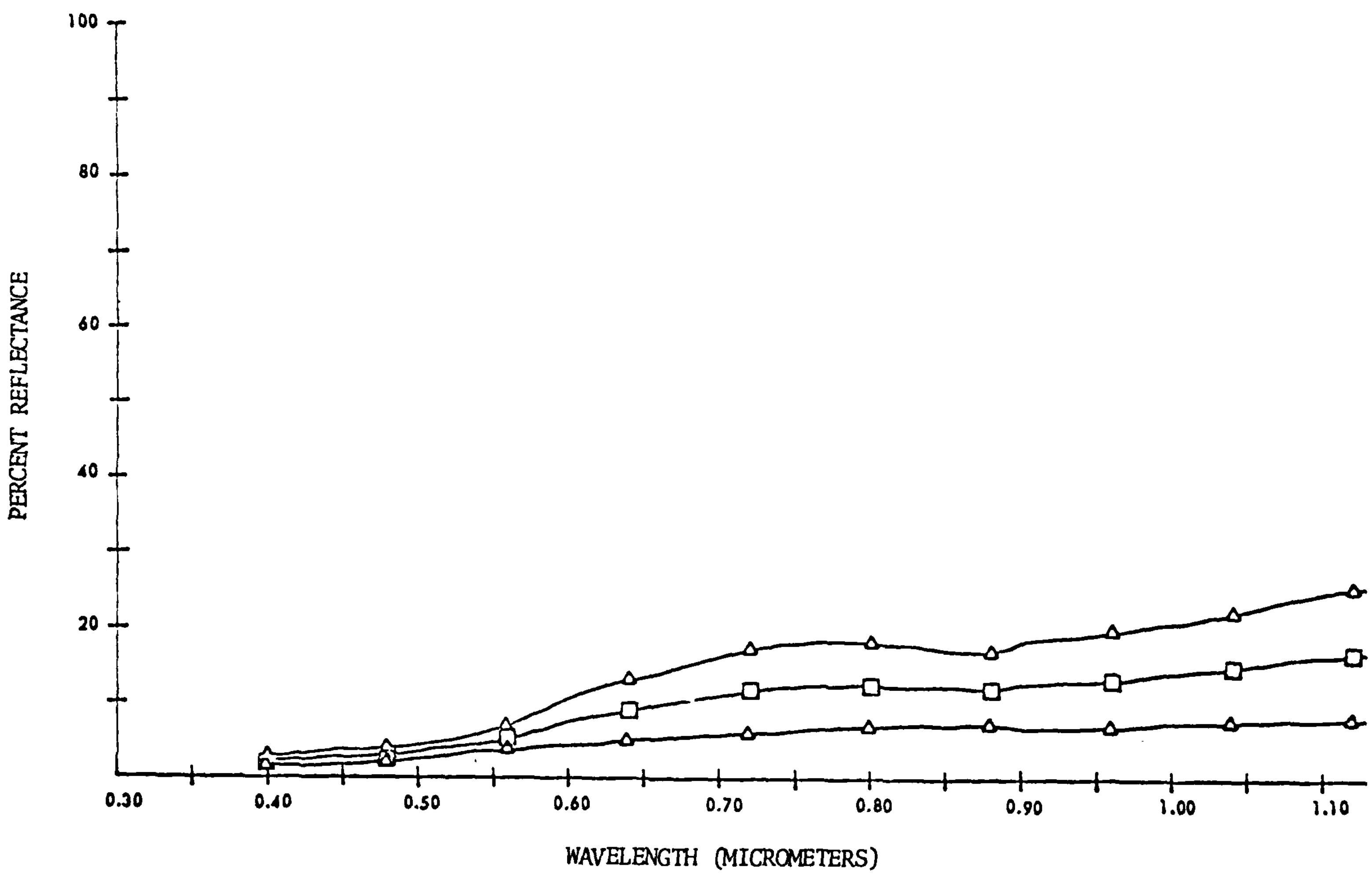


FIGURE C-6. WET LOAM

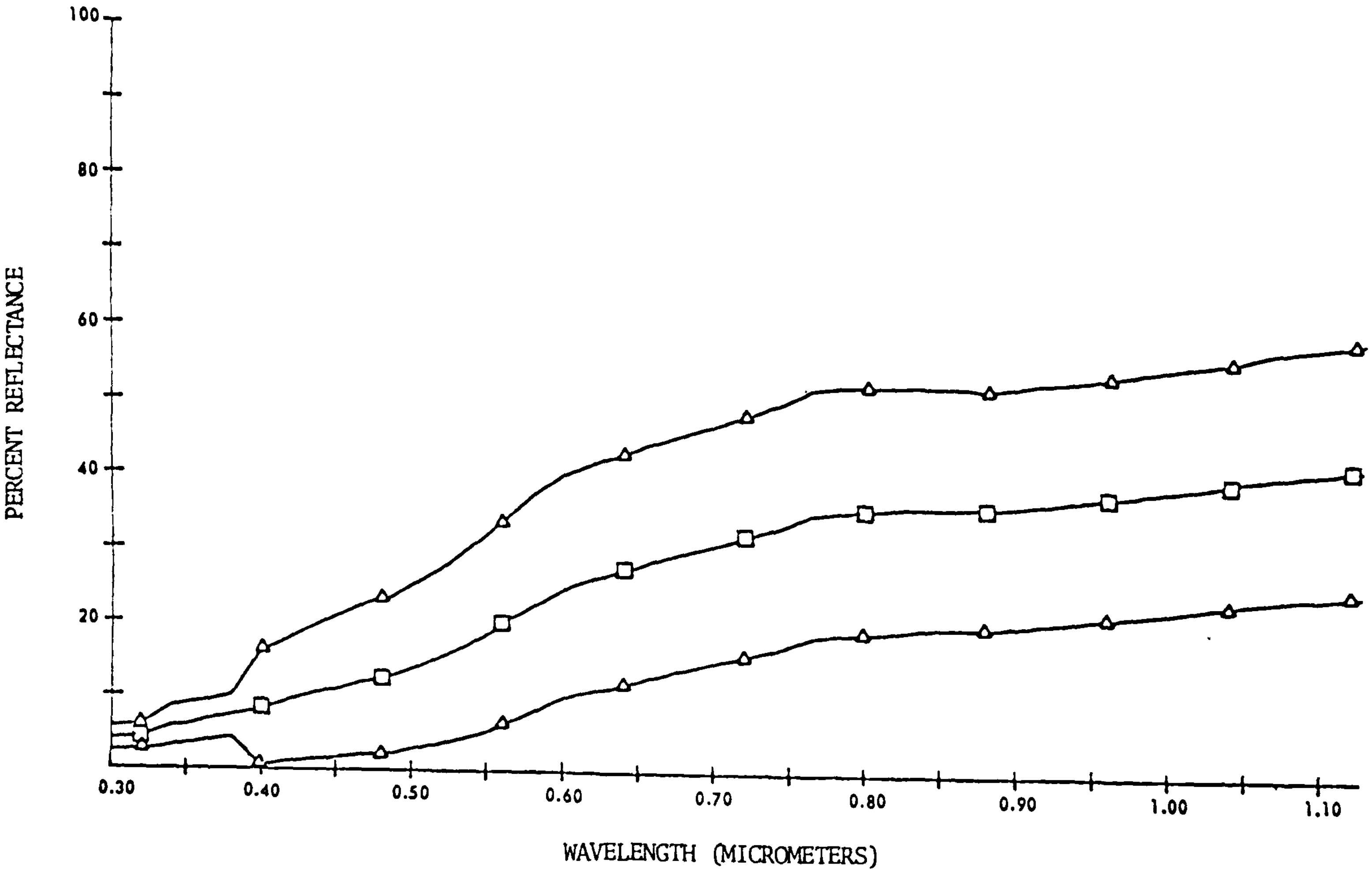


FIGURE C-7. ALL DRY SOILS

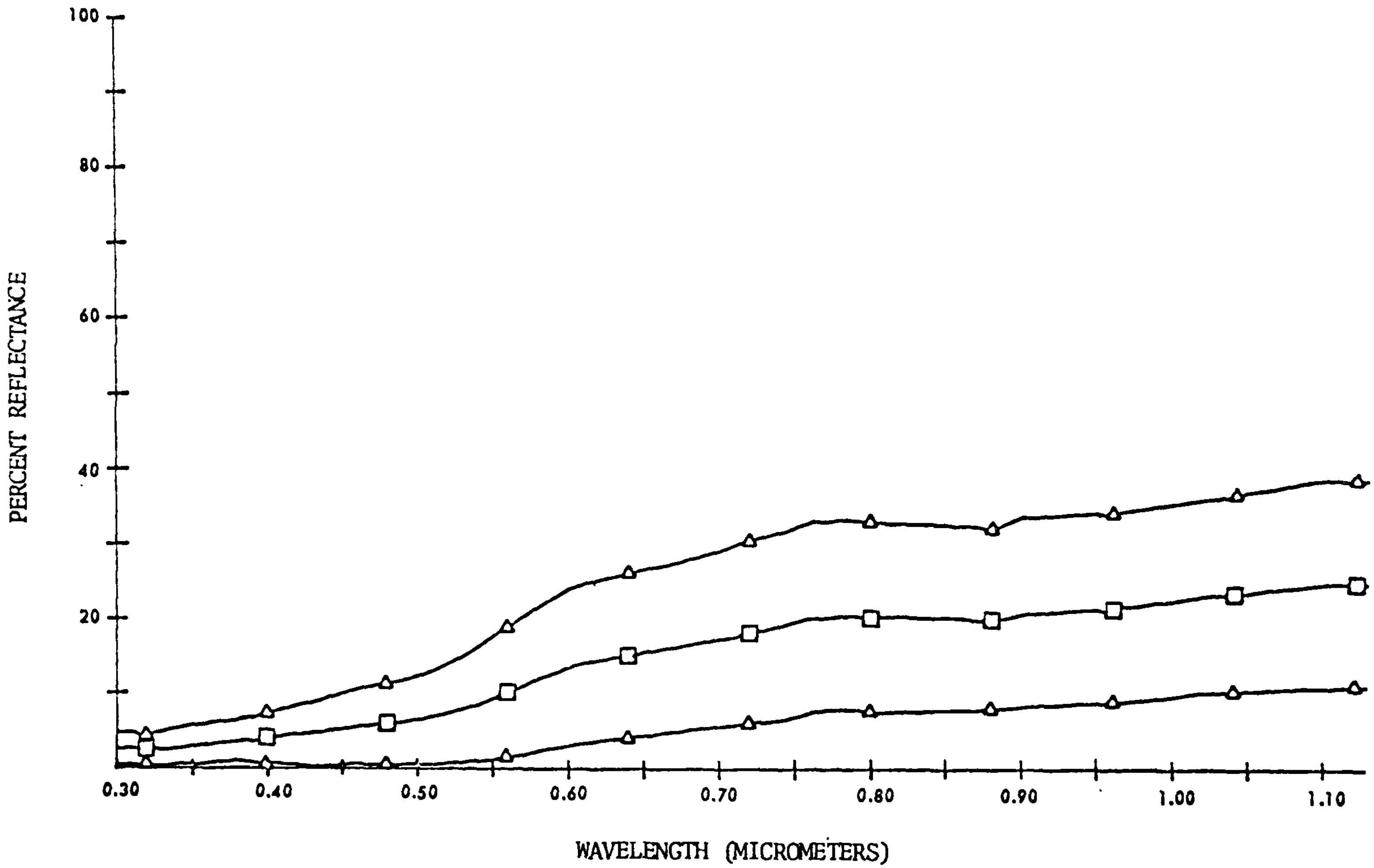


FIGURE C-8. ALL WET SOILS

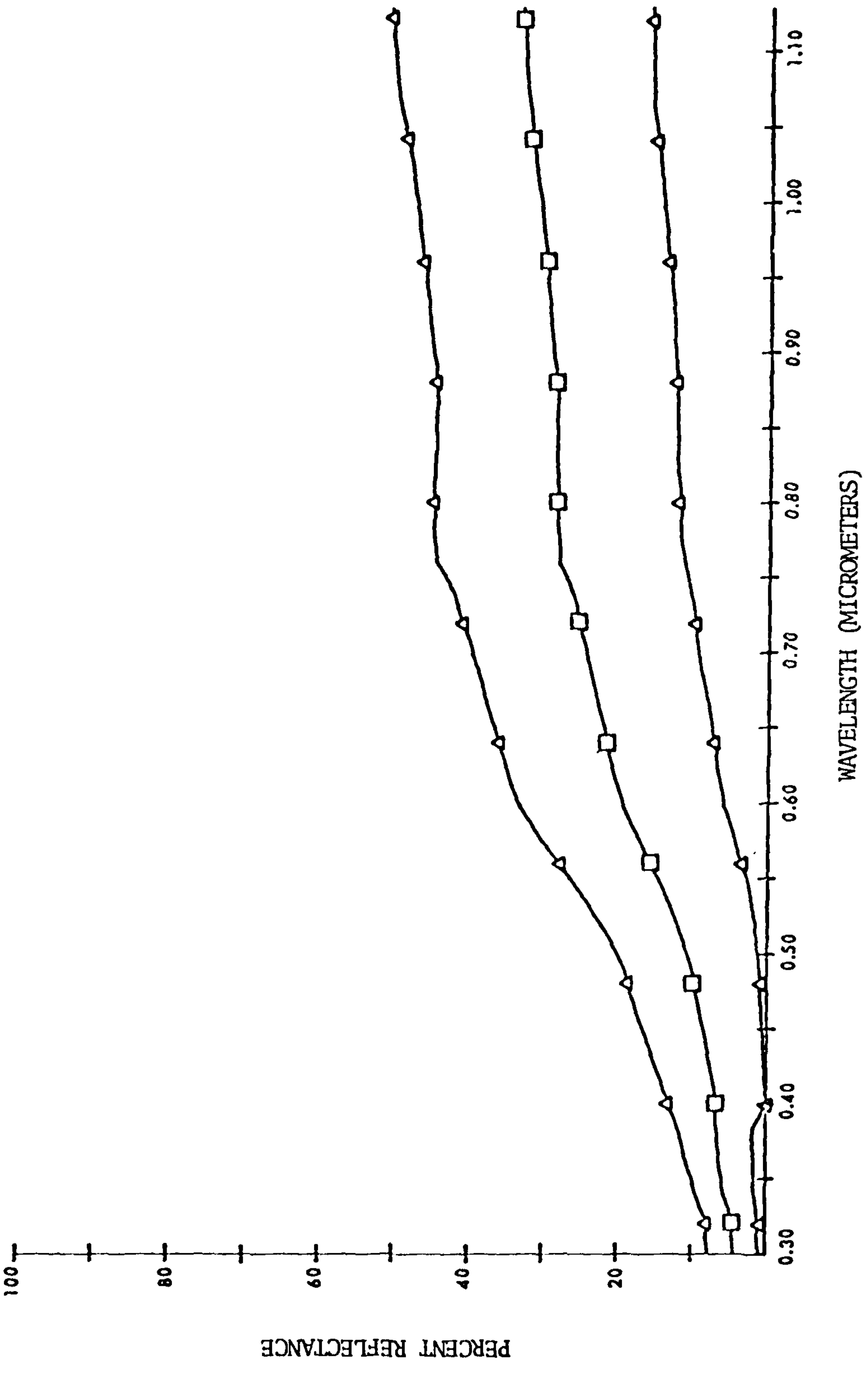


FIGURE C-9. ALL SOILS

indicate the percentage saturation of the soils when reflectance was measured, except that they were either "wet" or "dry."

There are two curves on cement/concrete. One in Figure C-10 is a composite of concrete and cement containing several different samples (14 curves), while the other in Figure C-11 is concrete sample (4 curves) from Willow Run Airport (runway and apron areas) taken after it had weathered 24 years. A comparison of these two sets will show that the concrete set from Willow Run Airport, although similar in shape of curves, is lower in the percentage of reflectance throughout the range under consideration, in comparison in the first composite curve. Although it is difficult to state specifically what the causes are that combine to make this difference, it is felt that one of the contributing factors is that the amount of rubber, oil, and grease rubbed into the surfaces over a long period of time would definitely lower the reflectance in the visible portion of the spectrum. In addition, the percentage of each concrete ingredient may make some difference, depending upon the composition for different types of application.

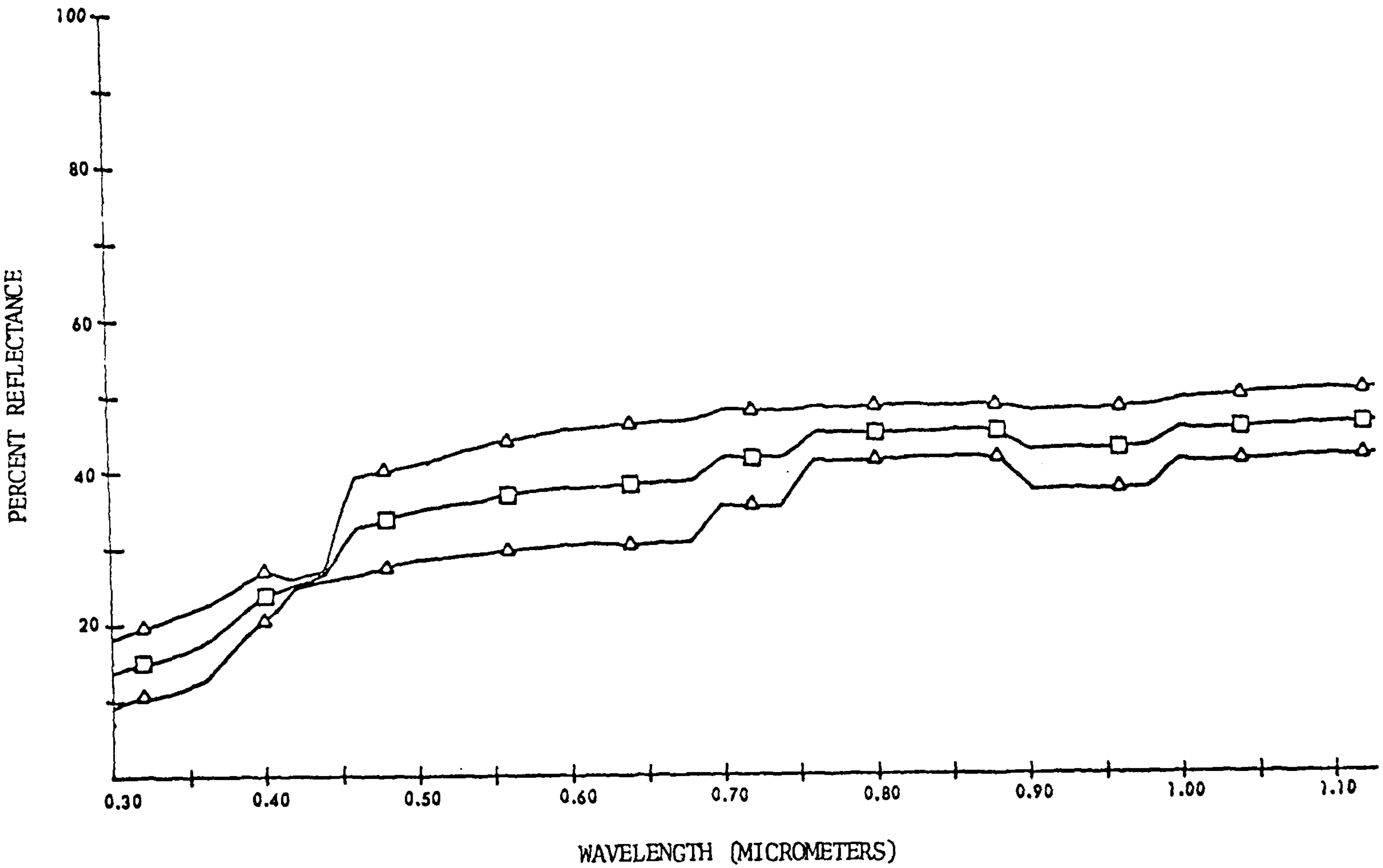


FIGURE C-10. NEW CONCRETE

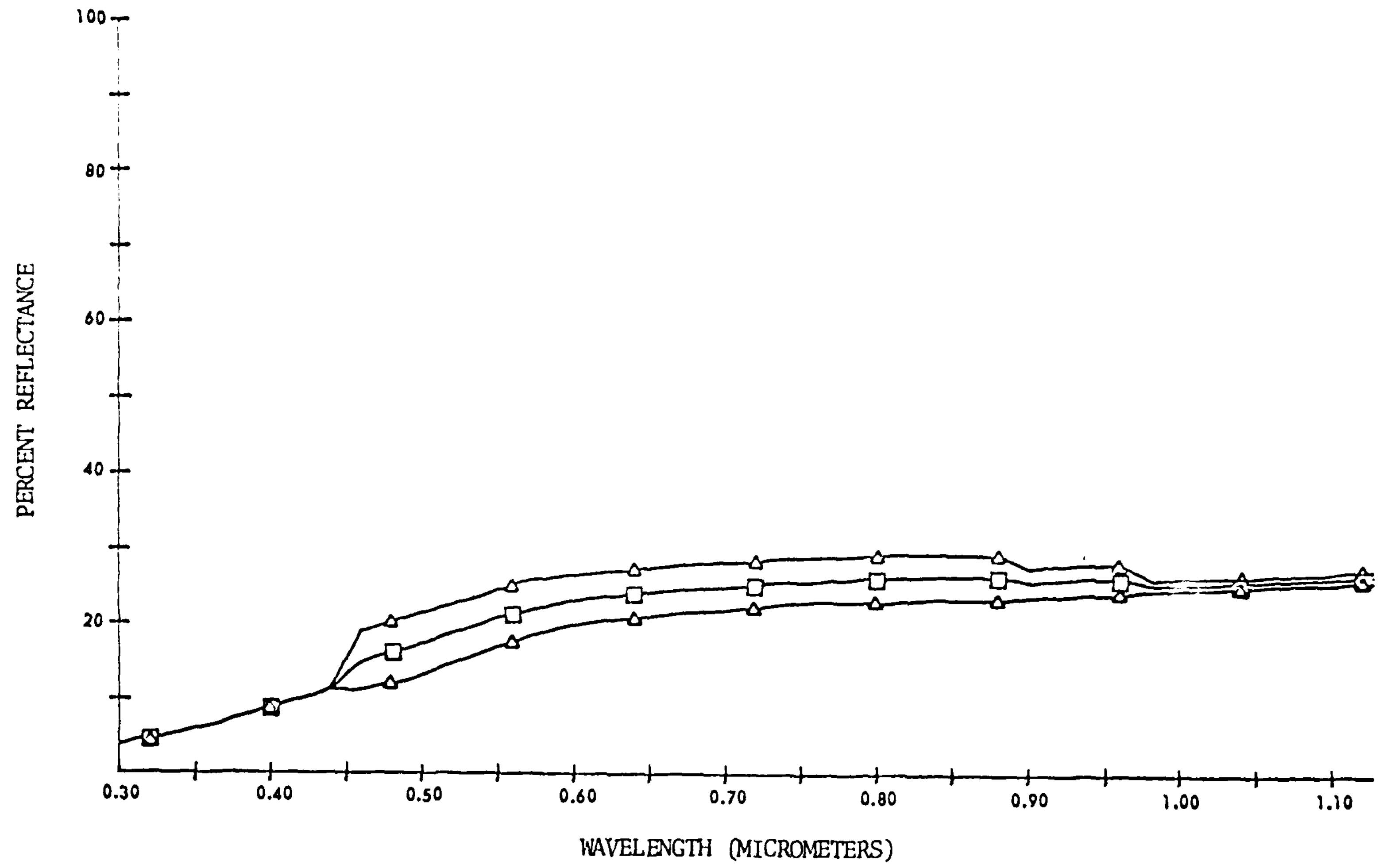


FIGURE C-11. WEATHERED CONCRETE

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

DATE
FILMED

0-8